**জাভাস্ক্রিপ্ট এইচটিএমএল কন্টেন্ট পরিবর্তন করতে পারে**

জাভাস্ক্রিপ্টে বহুবিধ মেথড রয়েছে তন্মধ্যে **getElementById()** মেথডটি অন্যতম।

এই উদাহরনে **getElementById()** মেথডটি id="test" যুক্ত এইচটিএমএল এলিমেন্টকে খুঁজে বের করার জন্য ব্যবহার করা হয়েছে এবং এলিমেন্টের কন্টেন্ট "হ্যালো জাভাস্ক্রিপ্ট" দ্বারা পরিবর্তন করা হয়েছে।

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>জাভাস্ক্রিপ্ট উদাহরণ</h3>

<p id="test">জাভাস্ক্রিপ্ট এইচটিএলএল কন্টেন্ট পরিবর্তন করতে পারে।</p>

<button type="button"

onclick="document.getElementById('test').innerHTML = 'হ্যালো জাভাস্ক্রিপ্ট!'">ক্লিক করুন</button>

</body>

</html>

**ফলাফল**

### জাভাস্ক্রিপ্ট উদাহরণ

হ্যালো জাভাস্ক্রিপ্ট!

**জাভাস্ক্রিপ্ট এইচটিএমএল এট্রিবিউট পরিবর্তন করতে পারে**

এই উদাহরণটিতে **<input>** ট্যাগের **type** এট্রিবিউট পরিবর্তন করা হয়েছেঃ

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>জাভাস্ক্রিপ্ট উদাহরণ</h3>

<button onclick="document.getElementById('test').type='text">পাসওয়ার্ড শো</button><br><br>

<input id="test" type="password" value="123456"><br><br>

<button onclick="document.getElementById('test').type='password'">পাসওয়ার্ড হাইড</button>

</body>

</html>

**ফলাফল**

**জাভাস্ক্রিপ্ট এইচটিএমএল স্টাইল পরিবর্তন করতে পারে**

এইচটিএমএল এলিমেন্টের এট্রিবিউট পরিবর্তন করা থেকে এইচটিএমএল স্টাইল পরিবর্তন করা একটু ভিন্নঃ

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>জাভাস্ক্রিপ্ট উদাহরণ</h3>

<p id="test">জাভাস্ক্রিপ্ট এইচটিএমএল এলিমেন্টের স্টাইল পরিবর্তন করতে পারে।

<button type="button" onclick="document.getElementById('test').style.color='green'">ক্লিক করি!</button>

</body>

</html>

# জাভাস্ক্রিপ্ট কোথায় রাখবেন?

[« পূর্ববর্তী](https://www.sattacademy.com/js/js_intro.php)

[পরবর্তী »](https://www.sattacademy.com/js/js_syntax.php)

এইচটিএমএল পেজের মধ্যে <head> অথবা <body> অথবা উভয় ট্যাগ এর মধ্যে জাভাস্ক্রিপ্টকে রাখা যায়।

<script> ট্যাগ এর মধ্যে জাভাস্ক্রিপ্ট কোড লিখতে হয়।

## <script> ট্যাগ

আপনি যদি এইচটিএমএল এর মধ্যে জাভাস্ক্রিপ্ট কোড ব্যবহার করতে চান তাহলে জাভাস্ক্রিপ্ট কোডকে অবশ্যই <script> এবং </script> ট্যাগের মধ্যে লিখতে হবে।

**উদাহরণ**

<script>

document.getElementById("test").innerHTML = "আমার প্রথম জাভাস্ক্রিপ্ট";

</script>

Copy

জাভাস্ক্রিপ্ট এর পুরাতন ভার্সন সমূহে type এট্রিবিউট ব্যবহার করা হত। যেমনঃ <script type="text/javascript">।  
কিন্তু বর্তমানে আর এটার প্রয়োজন নাই। কারণ জাভাস্ক্রিপ্ট এইচটিএমএলের পূর্বনির্ধারিত স্ক্রিপ্টিং ল্যাংগুয়েজ।

## জাভাস্ক্রিপ্ট ফাংশন এবং ইভেন্ট

জাভাস্ক্রিপ্ট ফাংশন হচ্ছে একগুচ্ছ জাভাস্ক্রিপ্ট কোড যাকে আপনি "কল" করলে সম্পাদিত হবে।

উদাহরণস্বরূপঃ কোন ইভেন্ট(ঘটনা) ঘটলেই একটি ফাংশন সম্পাদিত হবে। যেমনঃ ব্যবহারকারী যদি মাউস ক্লিক করে।

পরবর্তীতে [ফাংশন](https://www.sattacademy.com/js/js_functions.php) এবং [ইভেন্ট](https://www.sattacademy.com/js/js_events.php) সম্পর্কে বিস্তারিত আলোচনা করা হয়েছে।

## <head> অথবা <body> এর মধ্যে জাভাস্ক্রিপ্ট

এইচটিএমএল ডকুমেন্টের মধ্যে আপনি যত খুশি জাভাস্ক্রিপ্ট কোড লিখতে/রাখতে পারবেন।

জাভাস্ক্রিপ্টকে এইচটিএমএলের <body> অথবা <head> অথবা উভয়ের মধ্যেই রাখা যেতে পারে।

|  |  |
| --- | --- |
|  | সবচেয়ে ভাল অভ্যাস হচ্ছে সকল কোড একই স্থানে রাখা। |

## <head>এলিমেন্টে জাভাস্ক্রিপ্ট

এই উদাহরণে জাভাস্ক্রিপ্ট ফাংশনকে এইচটিএমএল পেজের <head> অংশে রাখা হয়েছে।

বাটনে ক্লিক করলে ফাংশনটি আপনার ডাকে সাড়া দিবেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="bn">

<head>

<meta charset="utf-8">

<title> Head সেকশনে জাভাস্ক্রিপ্ট এর ব্যবহার</title>

<script>

function myFunc() {

document.getElementById("test").innerHTML = "প্যারাগ্রাফটি পরিবর্তিত হয়েছে ";

}

</script>

</head>

<body>

<h4>হেড সেকশনে জাভাস্ক্রিপ্ট কোড লেখা হয়েছে</h4>

<p id="test">সাধারণ প্যারাগ্রাফ </p>

<button type="button" onclick="myFunc()">

চেষ্টা করি</button>

</body>

</html>

### ফলাফল

## <body> এর মধ্যে জাভাস্ক্রিপ্ট

এই উদাহরণে জাভাস্ক্রিপ্ট ফাংশনকে এইচটিএমএল পেজের <body> এলিমেন্টে রাখা হয়েছে

বাটনে ক্লিক করলে ফাংশনটি আপনার ডাকে সাড়া দিবেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="bn">

<head>

<meta charset="utf-8">

<title> Head সেকশনে জাভাস্ক্রিপ্ট এর ব্যবহার</title>

</head>

<body>

<p id="test">সাধারণ প্যারাগ্রাফ </p>

<button type="button" onclick="myFunc()">চেষ্টা করি</button>

<script>

function myFunc() {

document.getElementById("test").innerHTML = "প্যারাগ্রাফটি পরিবর্তিত হয়েছে ";

}

</script>

</body>

</html>

Copy

### ফলাফল

|  |  |
| --- | --- |
|  | জাভাস্ক্রিপ্টকে <body> এলিমেন্টের নিচের দিকে রাখা সবচেয়ে ভাল অভ্যাস। এতে আপনার পেজ দ্রুত লোড হয়। কারণ জাভাস্ক্রিপ্ট কম্পাইলার আপনার পেজকে ধীরগতি সম্পন্ন করে দিতে পারে। |

## বাহ্যিক জাভাস্ক্রিপ্ট

জাভাস্ক্রিপ্টকে বাহ্যিক(external) ফাইলের মধ্যেও রাখা যায়ঃ

### script.js

function myFunc() {

document.getElementById("test"

একই ধরণের স্ক্রিপ্ট কোড যখন একাধিক ওয়েব পেজে ব্যবহার করার প্রয়োজন হয় তখন এক্সটার্নাল জাভাস্ক্রিপ্ট ব্যবহার করতে হয়।

জাভাস্ক্রিপ্ট ফাইলের ফাইল এক্সটেনশন হচ্ছে .js।

এক্সটার্নাল জাভাস্ক্রিপ্ট ব্যবহার করার জন্য জাভাস্ক্রিপ্ট ফাইলটি <script> ট্যাগের src এট্রিবিউটের মধ্যে রাখুনঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="bn">

<head>

<meta charset="utf-8">

<title>বাহ্যিক জাভাস্ক্রিপ্ট এর উদাহরণ</title>

</head>

<body>

<h3>এক্সটার্নাল জাভাস্ক্রিপ্ট</h3>

<p id="test">সাধারণ প্যারাগ্রাফ</p>

<button type="button" onclick="myFunc()">ক্লিক করি</button>

<p><strong>নোট:</strong> "Script.js" এক্সটার্নাল ফাইলের মধ্যে myFunc জমা আছে।</p>

<script src="Script.js"></script>

</body>

</html>

Copy

### ফলাফল

এক্সটার্নাল জাভাস্ক্রিপ্ট রেফারেন্স/ফাইলকে আপনি আপনার ইচ্ছামত <head> অথবা <body> এর মধ্যে রাখতে পারেন।

রেফারেন্স কোডগুলো এমন আচরণ করে যেন এগুলোকে <script> ট্যাগের মধ্যেই রাখা হয়েছে।

|  |  |
| --- | --- |
|  | এক্সটার্নাল স্ক্রিপ্টের মধ্যে <script> ট্যাগ ব্যবহার করা যাবে না। |

## এক্সটার্নাল জাভাস্ক্রিপ্টের সুবিধা

জাভাস্ক্রিপ্টকে এক্সটার্নাল ফাইলের মধ্যে রাখলে কিছু বিশেষ বিশেষ সুবিধা পাওয়া যায়ঃ

* এটা এইচটিএমএল এবং জাভাস্ক্রিপ্ট কোডকে পৃথক করে।
* এইচটিএমএল এবং জাভাস্ক্রিপ্ট কোড অধিক পাঠযোগ্য হয় এবং খুব সহজেই রক্ষণাবেক্ষণ করা যায়।
* ক্যাশড(cached) জাভাস্ক্রিপ্ট ফাইল পেজ লোডের গতি বাড়িয়ে দেয়।

## জাভাস্ক্রিপ্ট প্রোগ্রাম

একটি কম্পিউটার প্রোগ্রাম একগুচ্ছ "ইন্সট্রাকশন/নির্দেশনার" তালিকা যা কম্পিউটারের মাধ্যমে সম্পাদিত হয়।

প্রোগ্রামিং-এ এই নির্দেশনাগুলোকে স্টেটমেন্ট বলা হয়।

জাভাস্ক্রিপ্ট একটি প্রোগ্রামিং ল্যাংগুয়েজ।

সেমিকোলন(;) এর মাধ্যমে জাভাস্ক্রিপ্ট স্টেটমেন্টকে আলাদা করা হয়।

### উদাহরণ

<!DOCTYPE html>

<html lang="bn">

<head>

<meta charset="utf-8">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>জাভাক্রিপ্ট স্টেটমেন্টগুলো সেমিকোলন দ্বারা আলাদা করা হয়।</h4>

<p id="test"></p>

<script>

a = 1;

b = 2;

c = a + b;

document.getElementById("test").innerHTML = c;

</script>

</body>

</html>

### ফলাফল

|  |  |
| --- | --- |
|  | এইচটিএমএলে জাভাস্ক্রিপ্ট প্রোগ্রামগুলো ব্রাউজারের মাধ্যমে সম্পাদিত হয়। |

## জাভাস্ক্রিপ্ট স্টেটমেন্ট

জাভাস্ক্রিপ্ট স্টেটমেন্ট সাধারণত ভ্যালু, অপারেটর, এক্সপ্রেশন, কিওয়ার্ড এবং কমেন্ট নিয়ে তৈরি হয়।

## জাভাস্ক্রিপ্ট ভ্যালু

জাভাস্ক্রিপ্টে দুই ধরণের ভ্যালু ব্যবহার করা হয়। যথা- ফিক্সড ভ্যালু এবং ভ্যারিয়েবল ভ্যালু।

ফিক্সড ভ্যালুকে লিটারাল বলা হয়। ভ্যারিয়েবল ভ্যালুকে চলক বা ভ্যারিয়েবল বলা হয়।

## জাভাস্ক্রিপ্ট লিটারাল

ফিক্সড ভ্যালু লেখার সবচেয়ে গুরুত্বপূর্ণ নিয়ম হলোঃ

সংখ্যাকে পূর্ণ সংখ্যা অথবা দশমিকে লেখাঃ

var a = 105.24; // দশমিকসহ নম্বর

var b = 105; // দশমিক ছাড়া নম্বর

Copy

স্ট্রিং হচ্ছে টেক্সট যাকে সিঙ্গেল অথবা ডাবল কোটেশনের(উদ্ধৃতি চিহ্ন) মধ্যে লেখা হয়ঃ

var a = "Tamim";

var b = 'Tamim';

Copy

## জাভাস্ক্রিপ্ট ভ্যারিয়েবল

প্রোগ্রামিং ভাষায় ভ্যালু সংরক্ষন করার কাজে ভ্যারিয়েবল ব্যবহার করা হয়।

ভ্যারিয়েবল ডিক্লেয়ার করার সময় জাভাস্ক্রিপ্টে var কিওয়ার্ড ব্যবহার করা হয়।

ভ্যারিয়েবলের মধ্যে মান রাখার জন্য সমান(=) চিহ্ন ব্যবহার করা হয়।

নিচের উদাহরণে a কে প্রথমে var কিওয়ার্ডের মাধ্যমে ভ্যারিয়েবল হিসেবে ডিক্লেয়ার করা হয়, তারপর সমান চিহ্ন ব্যবহার করে এর মধ্যে 10 কে ভ্যালু হিসেবে রাখা হয়েছে।

### উদাহরণ

<!DOCTYPE html>

<html lang="bn">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>জাভাস্ক্রিপ্ট চলক</h3>

<h5>এই উদাহরনটিতে x কে চলক হিসেবে ধরা হয়েছে, পরে x এর মান ৬ নির্ধারন করা হয়েছে।</h5>

<p id="test"></p>

<script>

var a;

a = 10;

document.getElementById("test").innerHTML = a;

</script>

</body>

</html>

Copy

### ফলাফল

## জাভাস্ক্রিপ্ট অপারেটর

ভ্যারিয়েবলের মধ্যে মান রাখার জন্য জাভাস্ক্রিপ্টে সাধারণত এসাইনমেন্ট অপারেটর( = ) ব্যবহার করা হয়ঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="bn">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>= অপারেটর</h3>

<p id="test"></p>

<script>

var a = 32;

document.getElementById("test").innerHTML = a;

</script>

</body>

</html>

Copy

### ফলাফল

মান নির্ণয় করার জন্য জাভাস্ক্রিপ্টে সাধারণত গাণিতিক অপারেটর( + - \*  / ) ব্যবহার করা হয়ঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="bn">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>যোগ করার আগে গুন করার উপর জোর দেওয়া হয়।</h4>

<h5>কিন্তু গুন করার আগে প্রথম বন্ধনীর কাজ করা হয়।</h5>

<p id="test"></p>

<script>

document.getElementById("test").innerHTML = (14 + 16) \* 3;

</script>

</body>

</html>

Copy

### ফলাফল

## জাভাস্ক্রিপ্ট এক্সপ্রেশন

কিছু ভ্যালু, চলক এবং অপারেটরের সমন্বয়ে জাভাস্ক্রিপ্ট এক্সপ্রেশন গঠিত হয় যা একটি ভ্যালু নির্ণয় করে।

উদাহরণস্বরূপ, ৫ \* ১০ এর গুণন ৫০ এ রূপান্তরিত হয়।

### উদাহরণ

<!DOCTYPE html>

<html lang="bn">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>জাভাস্ক্রিপ্ট এক্সপ্রেশন</h3>

<h5>এক্সপ্রেশনের মাধ্যমে ভ্যালু নির্ণয়।</h5>

<p id="test"></p>

<script>

document.getElementById("test").innerHTML = 12 + 13;

</script>

</body>

</html>

Copy

### ফলাফল

অনেক সময় এক্সপ্রেশনের মধ্যে ভ্যালু হিসেবে ভ্যারিয়েবলও থাকতে পারেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="bn">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>জাভাস্ক্রিপ্ট এক্সপ্রেশন</h3>

<h5>এক্সপ্রেশনের মাধ্যমে ভ্যালু নির্ণয়।</h5>

<p id="test"></p>

<script>

var a = 12;

document.getElementById("test").innerHTML = a + 13;

</script>

</body>

</html>

Copy

### ফলাফল

জাভাস্ক্রিপ্টে মানগুলো বিভিন্ন টাইপের হতে পারে। যেমনঃ নাম্বার এবং স্ট্রিং।

উদাহরণস্বরূপ, "সাহীদ" + " " + "মাহমুদ", যোগ হয়ে "সাহীদ মাহমুদ" হয়ঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="bn">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>জাভাস্ক্রিপ্ট এক্সপ্রেশন</h3>

<h5>এক্সপ্রেশনের মাধ্যমে ভ্যালু নির্ণয়।</h5>

<p id="test"></p>

<script>

document.getElementById("test").innerHTML = "সাহীদ" + " " + "মাহমুদ";

</script>

</body>

</html>

Copy

### ফলাফল

## জাভাস্ক্রিপ্ট কিওয়ার্ড

বিভিন্ন কার্যসম্পাদন করার জন্য জাভাস্ক্রিপ্ট কিওয়ার্ড ব্যবহার করা হয়।

যেমন- var কিওয়ার্ড ব্রাউজারকে ভ্যারিয়েবল(variable) তৈরি করতে বলেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="bn">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>var কীওর্য়াড ভেরিয়েবল তৈরী করে।</h1>

<p id="test"></p>

<script>

var a = 2 + 5;

var b = a \* 12;

document.getElementById("test").innerHTML = b;

</script>

</body>

</html>

Copy

### ফলাফল

## জাভাস্ক্রিপ্ট কমেন্ট

জাভাস্ক্রিপ্টের সকল স্টেটমেন্ট সম্পাদিত হয়না।

ডাবল স্ল্যাস(//)এর পরের অথবা /\* এবং \*/ এর মাঝের কোডগুলোকে কমেন্ট হিসাবে গণ্য করা হয়।

কমেন্টের কোডগুলোকে উপেক্ষা করা হয় এবং এগুলো সম্পাদিত হবে নাঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="bn">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>কমেন্ট সম্পাদিত হয় না।</h3>

<p id="test"></p>

<script>

var a = 10;

// var a = 23; ইহা সম্পাদিত হবে না

document.getElementById("test").innerHTML = a;

</script>

</body>

</html>

Copy

### ফলাফল

## জাভাস্ক্রিপ্ট আইডেন্টিফায়ার

আইডেন্টিফায়ারসমূহ হচ্ছে নাম

জাভাস্ক্রিপ্টে ভ্যারিয়েবল,কীওয়ার্ড এবং ফাংশনের নাম দেওয়ার জন্য আইডেন্টিফায়ার ব্যবহার করা হয়।

অধিকাংশ প্রোগ্রামেই নামকরনের নিয়ম একই থাকে।

জাভাস্ক্রিপ্টে প্রথম ক্যারেক্টারটি অবশ্যই অক্ষর, আন্ডারস্কোর(\_) অথবা ডলার($) চিহ্ন হবে।

পরের ক্যারেক্টারগুলো অক্ষর(characters), সংখ্যা, আন্ডারস্কোর(\_) অথবা ডলার($) চিহ্ন হতে পারে।

|  |  |
| --- | --- |
| **Note** | প্রথম অক্ষরটি কখনো সংখ্যা হবে না। এই পদ্ধতিতে জাভাস্ক্রিপ্ট খুব সহজেই নাম্বার থেকে আইডেন্টিফায়ারকে পৃথক করতে পারে। |

## জাভাস্ক্রিপ্ট কেস সেনসিটিভ

সকল জাভাস্ক্রিপ্ট আইডেন্টিফায়ার কেস-সেনসিটিভ(case-sensitive)।

lastName এবং lastname এই দুটি ভ্যারিয়েবল সম্পূর্ণ আলাদা

### উদাহরণ

<!DOCTYPE html>

<html lang="bn">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>জাভাস্ক্রিপ্ট কেস-সেন্সেটিভ</h3>

<h5>lastName কে lastname এ পরিবর্তন করার চেষ্টা করুন।</h5>

<p id="test"></p>

<script>

var lastName = "Tamim";

var lastname = "Tahmid";

document.getElementById("test").innerHTML = lastName;

</script>

</body>

</html>

Copy

### ফলাফল

জাভাস্ক্রিপ্ট VAR অথবা Var কে var কিওয়ার্ড হিসাবে বিবেচনা করে না।

## জাভাস্ক্রিপ্ট ক্যামেল কেস

প্রোগ্রামাররা একের অধিক শব্দকে এক শব্দে লেখার জন্য সচারচর তিনটি পদ্ধতি ব্যবহার করেঃ

হাইফেন(-)

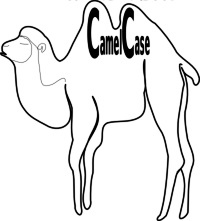
first-name, last-name, master-card, inter-city

আন্ডারস্কোর(\_)

first\_name, last\_name, master\_card, inter\_city

ক্যামেল কেস

FirstName, LastName, MasterCard, InterCity



প্রোগ্রামিং ল্যাংগুয়েজে বিশেষত জাভাস্ক্রিপ্টে ক্যামেল কেস ছোট হাতের অক্ষর দিয়ে শুরু হয়ঃ

firstName, lastName, masterCard, interCity.

|  |  |
| --- | --- |
| **Note** | জাভাস্ক্রিপ্টে হাইফেন(-) ব্যবহার করা যায় না। জাভাস্ক্রিপ্ট হাইফেনকে বিয়োগ চিহ্ন হিসাবে বিবেচনা করে। |

## জাভাস্ক্রিপ্ট ক্যারেক্টার সেট

জাভাস্ক্রিপ্ট ইউনিকোড ক্যারেক্টার সেট ব্যবহার করে।

ইউনিকোডে প্রায় সকল ক্যারেক্টার, সিম্বল এবং পাংচুয়েশন থাকে।

**জাভাস্ক্রিপ্ট প্রদর্শনের পদ্ধতি**

জাভাস্ক্রিপ্ট বিভিন্ন পদ্ধতিতে তথ্যকে প্রদর্শন করতে পারেঃ

* **window.alert()** ব্যবহার করে এলার্ট বক্সের মাধ্যমে
* **document.write()** ব্যবহার করে এইচটিএমএল ডকুমেন্টে লিখে
* **.innerHTML** ব্যবহার করে এইচটিএলএল এলিমেন্টে লিখে
* **console.log()** ব্যবহার করে ব্রাউজার কনসোলের মধ্যে দেখানো যায়

**window.alert() ব্যবহার**

তথ্য প্রদর্শনীর জন্য আপনি এলার্ট বক্স ব্যবহার করতে পারবেনঃ

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>আমার প্রথম ওয়েব পেইজ</h3>

<h5>আমার প্রথম অনুচ্ছেদ</h5>

<script>

window.alert("স্বাগতম!");

</script>

</body>

</html>

Copy

**ফলাফল**

**document.write() ব্যবহার**

পরীক্ষা করার উদ্দ্যেশে **document.write()** ব্যবহার করা সুবিধাজনকঃ

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>আমার প্রথম ওয়েব পেইজ</h3>

<h5>আমার প্রথম অনুচ্ছেদ</h5>

<script>

document.write("স্বাগতম!");

</script>

</body>

</html>

Copy

**ফলাফল**

এইচটিএমএল এলিমেন্ট সম্পূর্ন লোড হওয়ার পর document.write() ব্যবহার করলে আগের সকল এইচটিএমএলকে মুছে ফেলেঃ

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>আমার প্রথম ওয়েব পেইজ</h3>

<h5>আমার প্রথম অনুচ্ছেদ</h5>

<button onclick="document.write("স্বাগতম!")">ক্লিক কর</button>

</body>

</html>

Copy

**ফলাফল**

|  |  |
| --- | --- |
| **Note** | শুধুমাত্র পরীক্ষা করার জন্য document.write() পদ্ধতিটি ব্যবহার করা উচিত। |

**innerHTML ব্যবহার**

এইচটিএমএল এলিমেন্টকে এক্সেস করার জন্য জাভাস্ক্রিপ্ট **document.getElementById(id)** পদ্ধতিটি ব্যবহার করা হয়।

**id** এট্রিবিউটটি এইচটিএমএল এলিমেন্টকে নির্দেশ করে এবং **innerHTML** প্রোপার্টিটি এইচটিএমএল কন্টেন্টকে নির্দেশ করেঃ

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>আমার প্রথম ওয়েব পেইজ</h3>

<h5>আমার প্রথম অনুচ্ছেদ</h5>

স্বাগতম!

document.getElementById("test").innerHTML = "স্বাগতম!";

</body>

</html>

Copy

**ফলাফল**

|  |  |
| --- | --- |
| **Note** | এইচটিএমএল এলিমেন্টের innerHTML প্রোপার্টিকে পরিবর্তন করার মাধ্যমে এইচটিএমএলে ডাটা প্রদর্শন করা একটি সাধারণ পদ্ধতি। |

**console.log() ব্যবহার**

আপনার ব্রাউজার কনসোলে ডাটা ডিসপ্লের জন্য আপনি **console.log()** পদ্ধতি ব্যবহার করতে পারেন।

F12 এর মাধ্যমে ব্রাউজার কনসোল সক্রিয় করা হয় এবং মেনু থেকে "Console" নির্বাচন করি।

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>আমার প্রথম ওয়েব পেইজ</h3>

<h5>আমার প্রথম অনুচ্ছেদ</h5>

<script>

console.log("স্বাগতম!");

</script>

</body>

</html>

Copy

**ফলাফল**

স্ট্রিং(String), সংখ্যা(Number), বুলিয়ান(Boolean), অ্যারে(Array), অবজেক্ট(Object)।

## জাভাস্ক্রিপ্ট তথ্যের ধরন(Data Type)

জাভাস্ক্রিপ্ট ভ্যারিয়েবল অনেক টাইপের ডাটা জমা রাখতে পারেঃ সংখ্যা(number), স্ট্রিং(string), অ্যারে(array), অবজেক্ট(object) এবং আরো অনেকঃ

var scores = 16; // সংখ্যা

var name = "Tamim"; // স্ট্রিং

var animals = ["Tiger", "Lion", "Elephant"]; // অ্যারে

var a = {firstName:"Azizur", lastName:"Rahman"}; // অবজেক্ট

Copy

## ডাটা টাইপের ধারনা

প্রোগ্রামিং-এর ক্ষেত্রে, ডাটা টাইপ একটি গুরুত্বপূর্ন বিষয়।

ভ্যারিয়েবল নিয়ে কাজ করতে হলে ডাটা টাইপ জানা খুবই গুরুত্বপূর্ণ।

ডাটা টাইপ ছাড়া কম্পিউটার এগুলোকে(ডাটাকে) সঠিকভাবে সমাধান করতে পারেনাঃ

var a = 28 + "Tamim";

Copy

28 এর সাথে "Tamim" যোগ করা কি কোন অর্থ বহন করে? ইহা কি কোন ভুল(error) দেখাবে নাকি ফলাফল দিবে?

জাভাস্ক্রিপ্ট উপরের উদাহরণকে নিচের মত ভাববেঃ

var a = "28" + "Tamim";

Copy

|  |  |
| --- | --- |
| **Note** | যখন সংখ্যাকে স্ট্রিং-এর সাথে যোগ করা হয়, জাভাস্ক্রিপ্ট সংখ্যাকে স্ট্রিং হিসাবে গণ্য করে। |

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>যখন একটি সংখ্যা এবং একটি স্ট্রিং যুক্ত করা হয় তখন জাভাস্ক্রিপ্টের সংখ্যাটি স্ট্রিং হিসাবে গণ্য করে।

।</h4>

<p id="test"></p>

<script>

var a = 28 + "Tamim";

document.getElementById("test").innerHTML = a;

</script>

</body>

</html>

Copy

### ফলাফল

#### যখন একটি সংখ্যা এবং একটি স্ট্রিং যুক্ত করা হয় তখন জাভাস্ক্রিপ্টের সংখ্যাটি স্ট্রিং হিসাবে গণ্য করে। ।

28Tamim

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>যখন একটি সংখ্যা এবং একটি স্ট্রিং যুক্ত করা হয়,তখন জাভাস্ক্রিপ্ট সংখ্যাকে স্ট্রিং হিসাবে গণ্য করে। </h4>

<p id="test"></p>

<script>

var a = "Tamim" + 28;

document.getElementById("test").innerHTML = a;

</script>

</body>

</html>

Copy

### ফলাফল

#### যখন একটি সংখ্যা এবং একটি স্ট্রিং যুক্ত করা হয় তখন জাভাস্ক্রিপ্টের সংখ্যাটি স্ট্রিং হিসাবে গণ্য করে। ।

28Tamim

জাভাস্ক্রিপ্ট এক্সপ্রেশনকে বাম থেকে ডান দিকে সম্পাদন করে। ডাটা টাইপের ধারাবাহিকতা ভিন্ন ভিন্ন ফলাফল প্রদর্শন করতে পারেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var a = 28 + 12 + "Tamim";

document.getElementById("test").innerHTML = a;

</script>

</body>

</html>

Copy

### ফলাফল

40Tamim

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var a = "Tamim" + 28 + 12;

document.getElementById("test").innerHTML = a;

</script>

</body>

</html>

Copy

### ফলাফল

Tamim2812

প্রথম উদাহরণে, 28 এবং 12কে সংখ্যা হিসাবে বিবেচনা করে যোগ করে তারপর "Tamim" যেহেতু স্ট্রিং তাই জাভাস্ক্রিপ্ট সম্পূর্নটাকে স্ট্রিং বিবেচনা করে।

দ্বিতীয় উদাহরণে, প্রথম অপারেন্ড স্ট্রিং হওয়ায় সকল অপারেন্ডকেই স্ট্রিং হিসাবে গণ্য করা হয়েছে।

## জাভাস্ক্রিপ্ট ডায়নামিক ডাটা টাইপ

জাভাস্ক্রিপ্টে ডাটা টাইপ ডায়নামিক। এর অর্থ একই ভ্যারিয়েবলে বিভিন্ন টাইপের ডাটা রাখা যেতে পারেঃ

var a; // এখানে a হচ্ছে undefined

var a = 9; // এখানে a হচ্ছে সংখ্যা(Number)

var a = "Tamim"; // এখানে a হচ্ছে স্ট্রিং(String)

Copy

## জাভাস্ক্রিপ্ট স্ট্রিং(String)

স্ট্রিং(string) হচ্ছে কিছু অক্ষরের(character) ধারা। যেমন- "আজিজুর রহমান"।

স্ট্রিংকে উদ্ধৃতি(quote) চিহ্নের মধ্যে লিখতে হয়। আপনি সিঙ্গেল অথবা ডাবল কোটেশন ব্যবহার করতে পারেনঃ

var newCar = "Volvo XC60"; // ডাবল কোট ব্যবহার করে

var newCar = 'Volvo XC60'; //সিঙ্গেল কোট ব্যবহার করে

Copy

আপনি স্ট্রিং-এর মধ্যেও কোটেশন ব্যবহার করতে পারেন, তবে স্ট্রিং-এর উভয় পাশের কোটেশনের সাথে মিলতে পারবে নাঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var wel1 = "Welcome To Satt Academy"

var wel2 = "Welcome To 'Satt' Academy"

var wel3 = 'Welcome To "Satt" Academy'

document.getElementById("test").innerHTML =

wel1 + "<br>" + wel2 + "<br>" + wel3;

</script>

</body>

</html>

Copy

### ফলাফল

Welcome To Satt Academy  
Welcome To 'Satt' Academy  
Welcome To "Satt" Academy

এই টিউটোরিয়ালের পরবর্তী পরিচ্ছেদে স্ট্রিং সম্পর্কে আপনি আরো শিখবেন।

## জাভাস্ক্রিপ্ট সংখ্যা

জাভাস্ক্রিপ্টে শুধু এক ধরণের সংখ্যা আছে।

সংখ্যাকে দশমিক অথবা দশমিক ছাড়াও লেখা যায়ঃ

var a = 12.00; // দশমিক(decimals) যুক্ত সংখ্যা

var b = 12; // দশমিক ছাড়া সংখ্যা

Copy

অতিরিক্ত বড় অথবা অতিরিক্ত ছোট সংখ্যাকে বৈজ্ঞানিক (exponential) পদ্ধতিতে লেখা যেতে পারেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var a = 12.00;

var b = 12;

var c = 324e4;

var d = 324e-4;

document.getElementById("test").innerHTML = a + "<br>" + b + "<br>" + c + "<br>" + d

</script>

</body>

</html>

Copy

### ফলাফল

12  
12  
3240000  
0.0324

এই টিউটোরিয়ালের পরবর্তী পরিচ্ছেদে সংখ্যা সম্পর্কে আপনি আরো শিখবেন।

## জাভাস্ক্রিপ্ট বুলিয়ান(Boolean)

বুলিয়ানের শুধুমাত্র দুইটি মান থাকেঃ true অথবা false ।

var a = true;

var b = false;

Copy

শর্ত পরীক্ষা(Condition Test) করার জন্য প্রায়ই বুলিয়ান ব্যবহার করা হয়।

এই টিউটোরিয়ালের পরবর্তী পরিচ্ছেদে শর্ত পরীক্ষা(Condition Test) সম্পর্কে আপনি আরো অনেক কিছু শিখবেন।

## জাভাস্ক্রিপ্ট অ্যারে(Array)

জাভাস্ক্রিপ্ট অ্যারেকে তৃতীয় বন্ধনীর(Square Bracket) মাধ্যমে লিখতে হয়।

অ্যারের উপাদানগুলো(items) কমার(,) মাধ্যমে আলাদা করতে হয়।

নিচের উদাহরণে animals নামের একটি অ্যারে ভ্যারিয়েবল ডিক্লেয়ার করা হয়েছে, যার মধ্যে তিনটি মান রয়েছেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals[1];

</script>

</body>

</html>

Copy

### ফলাফল

Tiger

অ্যারের ইনডেক্স(index) শুরু হয় 0 দিয়ে, অর্থাৎ প্রথম উপাদানটি হবে [0],দ্বিতীয় উপাদানটি হবে [1], এবং এভাবে চলতে থাকবে।

এই টিউটোরিয়ালের পরবর্তী পরিচ্ছেদে অ্যারে(Array) সম্পর্কে আপনি আরো শিখবেন।

## জাভাস্ক্রিপ্ট অবজেক্ট(Object)

দ্বিতীয় বন্ধনীর(curly bracket) মাধ্যমে জাভাস্ক্রিপ্ট অবজেক্ট লেখা হয়।

অবজেক্টের প্রোপার্টিগুলো name:value জোড়া আকারে লিখতে হয়, একাধিক প্রোপার্টিকে কমার(,) মাধ্যমে পৃথক করতে হয়।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>একটি জাভাস্ক্রিপ্ট অবজেক্ট তৈরি </h4>

<p id="test"></p>

<script>

var member = {firstName:"Azizur", lastName:"Rahman", age:32, eyeColor:"black"};

document.getElementById("test").innerHTML =

member.firstName + " " + member.lastName + " " + "এর বয়স" + " " + member.age + " " + " বছর।";

</script>

</body>

</html>

Copy

### ফলাফল

#### একটি জাভাস্ক্রিপ্ট অবজেক্ট তৈরি

Azizur Rahman এর বয়স 32 বছর।

উপরের উদাহরণে person অবজেক্টের চারটি প্রোপার্টি আছেঃ firstName, lastName, age, and eyeColor

এই টিউটোরিয়ালের পরবর্তী পরিচ্ছেদে অবজেক্ট(Object) সম্পর্কে আপনি আরো শিখবেন।

## typeof অপারেটর

জাভাস্ক্রিপ্ট ভ্যারিয়েবলের টাইপ জানতে আপনি typeof অপারেটর(operator) ব্যবহার করতে পারেনঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> typeof অপারেটর দ্বারা ভ্যারিয়েবল, অবজেক্ট, ফাংশন, অথবা এক্সপ্রেশন এর টাইপকে বুঝায় ।</h3>

<p id="test"></p>

<script>

document.getElementById("test").innerHTML =

typeof "Tahmid" + "<br>" +

typeof 3.14 + "<br>" +

typeof NaN + "<br>" +

typeof false + "<br>" +

typeof new Date() + "<br>" +

typeof function () {} + "<br>" +

typeof newDay + "<br>" +

typeof [1,2,3,4] + "<br>" +

typeof {name:'আজিজুর', age:34} + "<br>" +

typeof null;

</script>

</body>

</html><

Copy

### ফলাফল

### typeof অপারেটর দ্বারা ভ্যারিয়েবল, অবজেক্ট, ফাংশন, অথবা এক্সপ্রেশন এর টাইপকে বুঝায় ।

string

number

number

boolean

object

function

undefined

object

object

object

|  |  |
| --- | --- |
| **Note** | জাভাস্ক্রিপ্টে অ্যারে হচ্ছে বিশেষ ধরণের অবজেক্ট(object)। তাই অ্যারের টাইপ অবজেক্ট(Object)। |

## অসংজ্ঞায়িত(Undefined)

জাভাস্ক্রিপ্টে ভ্যারিয়েবলকে ভ্যালু ছাড়া ডিক্লেয়ার করলে এর একটি ভ্যালু থাকে undefined। typeof এর মানও undefined ।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var member;

document.getElementById("test").innerHTML =

member + "<br>" + typeof member;

</script>

</body>

</html>

Copy

### ফলাফল

undefined  
undefined

ভ্যারিয়েবলের মান undefined সেট করে এর মান খালি রাখা যায়।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var member = undefined;

document.getElementById("test").innerHTML =

member + "<br>" + typeof member; // ভ্যালু undefined, টাইপও undefined

</script>

</body>

</html>

Copy

### ফলাফল

undefined  
undefined

## খালি মান(Empty Values)

খালি মান(value) আর অসজ্ঞায়িত দুইটি ভিন্ন জিনিস।

খালি স্ট্রিং-এর ভ্যালু এবং টাইপ(type) দুইটাই থাকে।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var member = "";

document.getElementById("test").innerHTML =

member + "<br>" + typeof member; // ভ্যারিয়েবলের ভ্যালু "", এবং typeof হচ্ছে string

</script>

</body>

</html>

Copy

### ফলাফল

string

## ফাকা(Null)

জাভাস্ক্রিপ্টে null দ্বারা বুঝায় "কিছুই না"। এটা দ্বারা বুঝায় যার কোন অস্তিত্ব নাই।

জাভাস্ক্রিপ্টে, null এর ডাটা টাইপ হচ্ছে অবজেক্ট(object)।

|  |  |
| --- | --- |
| **Note** | জাভাস্ক্রিপ্টে null এর typeof অপারেটরের ভ্যালু হচ্ছে অবজেক্ট, আপনি একে bug হিসেবে বিবেচনা করতে পারেন।কারণ এটা null হওয়া উচিৎ ছিল। |

আপনি অবজেক্টের মান null সেট করে একে খালি রাখতে পারেনঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var member = null;

document.getElementById("test").innerHTML =

member + "<br>" + typeof member; // মান হচ্ছে null, কিন্তু টাইপ এখনো অবজেক্ট

</script>

</body>

</html>

Copy

### ফলাফল

null  
object

## Undefined এবং Null এর মধ্যে পার্থক্য

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

document.getElementById("test").innerHTML =

typeof undefined + "<br>" +

typeof null + "<br>" +

(null === undefined) + "<br>" +

(null == undefined);

</script>

</body>

</html>

Copy

### ফলাফল

undefined  
object  
false  
true

**জাভাস্ক্রিপ্ট ভ্যারিয়েবল**

জাভাস্ক্রিপ্ট ভ্যারিয়েবল হচ্ছে ভ্যালু রাখার জন্য ব্যবহৃত কন্টেইনার।

নিচের উদাহরণে a, b এবং c হচ্ছে ভ্যারিয়েবলঃ

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>জাভাস্ক্রিপ্ট ভেরিয়েবল</h3>

<h4>এই উদাহরণে a, b, এবং c হলো ভেরিয়েবল</h4>

<p id="test"></p>

<script>

var a = 10;

var b = 20;

var c = a + b;

document.getElementById("test").innerHTML = c;

</script>

</body>

</html>

Copy

**ফলাফল**

### জাভাস্ক্রিপ্ট ভেরিয়েবল

#### এই উদাহরণে a, b, এবং c হলো ভেরিয়েবল

30

উপরের উদাহরনে লক্ষ্য করুন a, b এবং c নামের তিনটি ভ্যারিয়েবল আছেঃ

* a ভ্যারিয়েবলে সংরক্ষিত হয় 10
* b ভ্যারিয়েবলে সংরক্ষিত হয় 20
* c ভ্যারিয়েবলে সংরক্ষিত হয় 30 (10 + 20 = 30)

**ভ্যারিয়েবল প্রায় বীজগণিতের মতই**

উপরের উদাহরণে a, b এবং c হচ্ছে ভ্যারিয়েবলঃ

বীজগণিতের মত প্রোগ্রামিংয়ের ক্ষেত্রেও মান রাখার জন্য ভ্যারিয়েবল বা চলক (যেমন-a) ব্যবহার করা হয়।

বীজগণিতের মত প্রোগ্রামিংয়ের ক্ষেত্রেও আমরা এক্সপ্রেশনে ভ্যারিয়েবল ব্যবহার করি।  
যেমনঃ (c = a + b)

উপরের উদাহরণ থেকে হিসাব করলে যোগফল হয় 30

|  |  |
| --- | --- |
| **Note** | জাভাস্ক্রিপ্ট ভ্যারিয়েবল হচ্ছে ভ্যালু রাখার জন্য ব্যবহৃত কন্টেইনার। |

**জাভাস্ক্রিপ্ট আইডেন্টিফায়ার**

সকল জাভাস্ক্রিপ্ট **ভ্যারিয়েবল** অবশ্যই একটি **ইউনিক** নামের মাধ্যমে সনাক্ত করতে হবে।

এই ইউনিক নামগুলোকে **আইডেন্টিফায়ার** বলে।

আইডেন্টিফায়ারগুলোর সংক্ষিপ্ত নাম হতে পারে (যেমন- x এবং y),অথবা বর্ণনামূলক নামও হতে পারে (যেমন-age, sum, totalVolume)।

ভ্যারিয়েবলের নাম দেওয়ার ক্ষেত্রে সাধারণ নিয়মগুলো হচ্ছেঃ

* নামের মধ্যে অক্ষর(x, y, z), ডিজিট(1, 2, 3), আন্ডারস্কোর(\_) এবং ডলার($) চিহ্ন থাকতে পারে
* নাম অবশ্যই অক্ষর দিয়ে শুরু হবে
* $ এবং \_ দিয়েও নাম শুরু হতে পারে
* নামগুলো কেসসেন্সিটিভ (যেমন x এবং X দুটি আলাদা ভ্যারিয়েবল)
* সংরক্ষিত শব্দগুলোকে(যেমন- জাভাস্ক্রিপ্ট কিওয়ার্ড) কখনো নাম হিসাবে ব্যবহার করা যাবেনা

|  |  |
| --- | --- |
|  | জাভাস্ক্রিপ্ট আইডেন্টিফায়ারগুলো কেসসেন্সিটিভ। |

**এসাইনমেন্ট অপারেটর**

জাভাস্ক্রিপ্টে সমান চিহ্নকে (=) "এসাইনমেন্ট" অপারেটর হিসাবে ব্যবহার করা হয়, যা দ্বারা "সমান" বুঝায় না। বরং ভ্যারিয়েবলের মধ্যে ভ্যালু এসাইন করা বুঝায়।

কম্পিউটারের ভ্যারিয়েবল বীজগণিতের ভ্যারিয়েবল থেকে আলাদা। নিচের লাইনটি বীজগণিতে কোন অর্থ বহন করে নাঃ

a = b + c;

Copy

কিন্তু জাভাস্ক্রিপ্টে এর অর্থ আছে, এর মানে x + 10 এর মান x এর মধ্যে জমা হয়।

এটা x + 10 এর মানকে গণনা করে এবং ফলাফল x এর মধ্যে জমা রাখে। এখানে x এর মান 10 বৃদ্ধি পায়।

|  |  |
| --- | --- |
|  | জাভাস্ক্রিপ্টে "==" অপারেটরটি সমান বুঝাতে লেখা হয়। |

**জাভাস্ক্রিপ্ট ডাটাটাইপ**

জাভাস্ক্রিপ্ট ভ্যারিয়েবলের মধ্যে সংখ্যা(৪০),টেক্সট ভ্যালু ("আজিজুর রহমান") থাকতে পারে।

প্রোগ্রামিং-এ টেক্সট ভ্যালুকে টেক্সট স্ট্রিং বলে।

জাভাস্ক্রিপ্ট অনেক ধরণের তথ্য বা ডাটা পরিচালনা করতে পারে, কিন্তু এখন শুধু নাম্বার এবং স্ট্রিং শিখবো।

স্ট্রিংকে সিঙ্গেল বা ডাবল কোটেশনের(উদ্ধৃতির) মধ্যে লেখা হয়। সংখ্যাকে উদ্ধৃতি ছাড়াই লিখতে হয়।

যদি আপনি নাম্বারকে উদ্ধৃতির মধ্যে লিখেন তাহলে তা স্ট্রিং বলে গণ্য হবে।

var a = "Satt Academy";

var b = 'Satt Academy';

var c = 20;

var d = "20";

Copy

**জাভাস্ক্রিপ্ট ভ্যারিয়েবল তৈরি**

জাভাস্ক্রিপ্টে ভ্যারিয়েবল তৈরিকে অন্য ভাষায় ভ্যারিয়েবল **ডিক্লেয়ারেশন** বলে।

**var** কিওয়ার্ডের মাধ্যমে জাভাস্ক্রিপ্টে **ভ্যারিয়েবল** ডিক্লেয়ার করা হয়।

var academyName;

Copy

ঘোষণা করার পরে এখন ভ্যারিয়েবলের কোন মান নাই। প্রকৃতপক্ষে এটার মান **undefined**

ভ্যারিয়েবলের মধ্যে মান রাখার জন্য আমরা **সমান(=)** চিহ্ন ব্যবহার করি।

academyName = "Satt";

Copy

এমনকি ভ্যারিয়েবল ডিক্লেয়ারের সময়ই আপনি ভ্যারিয়েবলের মান দিয়ে দিতে পারেনঃ

var academyName = "Satt";

Copy

নিচের উদাহরণে আমরা academyName নামে ভ্যারিয়েবল ডিক্লেয়ার করে এর মান "Satt" দিয়েছি।

তারপরে আমরা id="test" সম্বলিত এইচটিএমএল এলিমেন্টের মধ্যে **আউটপুট** নিয়েছিঃ

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>জাভাস্ক্রিপ্ট পাত্রসমূহ</h3>

<h4>একটি ভেরিয়েবল তৈরী করুন,এর মান নির্দিষ্ট করুন, এবং এটি প্রদর্শন করান।</h4>

<p id="test"></p>

<script>

var academyName= "Satt";

document.getElementById("test").innerHTML = academyName;

</script>

</body>

</html>

Copy

**ফলাফল**

### জাভাস্ক্রিপ্ট পাত্রসমূহ

#### একটি ভেরিয়েবল তৈরী করুন,এর মান নির্দিষ্ট করুন, এবং এটি প্রদর্শন করান।

Satt

|  |  |
| --- | --- |
|  | সকল ভ্যারিয়েবলগুলোকে স্ক্রিপ্টের প্রথমেই ঘোষণা করা প্রোগ্রামিং-এ ভাল অভ্যাস। |

**একটি স্টেটমেন্টে অনেক ভ্যারিয়েবল**

একটি স্টেটমেন্টের মধ্যে আপনি অনেক ভ্যারিয়েবল ডিক্লেয়ার করতে পারেন।

**var** কিওয়ার্ডের মাধ্যমে শুরু করুন এবং **কমা(,)** দ্বারা ভ্যারিয়েবলগুলো আলাদা করুন, সবশেষে সেমিকোলন(;) দিনঃ

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>জাভাস্ক্রিপ্ট পাত্রসমূহ</h3>

<h4>আপনি একটি স্টেটমেন্টের মধ্যে অনেক ভেরিয়েবল(চলক) ঘোষণা করতে পারেন।</h4>

<p id="test"></p>

<script>

var person = "আজিজুর রাহমান", academyName = "satt", cost =" ফ্রি";

document.getElementById("test").innerHTML = academyName ;

</script>

</body>

</html>

Copy

**ফলাফল**

### জাভাস্ক্রিপ্ট পাত্রসমূহ

#### আপনি একটি স্টেটমেন্টের মধ্যে অনেক ভেরিয়েবল(চলক) ঘোষণা করতে পারেন।

satt

একের অধিক লাইনেও ডিক্লেয়ার করা যেতে পারেঃ

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>জাভাস্ক্রিপ্ট পাত্রসমূহ</h3>

<h4>আপনি একটি স্টেটমেন্টের মধ্যে অনেক ভেরিয়েবল(চলক) ঘোষণা করতে পারেন।</h4>

<p id="test"></p>

<script>

var person = "আজিজুর রাহমান",

academyName = "satt",

cost =" ফ্রি";

document.getElementById("test").innerHTML = academyName ;

</script>

</body>

</html>

Copy

**ফলাফল**

### জাভাস্ক্রিপ্ট পাত্রসমূহ

#### আপনি একটি স্টেটমেন্টের মধ্যে অনেক ভেরিয়েবল(চলক) ঘোষণা করতে পারেন।

satt

**Value = undefined**

কম্পিউটার প্রোগ্রামিং-এ, প্রায়ই ভ্যারিয়েবলকে মান ছাড়াই ডিক্লেয়ার করা হয়, মানগুলো হয় পরে নির্ধারন করা হয় অথবা হিসাব করে পাওয়া যায়।

যখন ভ্যারিয়েবলকে মান ছাড়াই ডিক্লেয়ার করা হয় তখন এর মান হয় **undefined**।

নিচের স্টেটমেন্টকে সম্পাদন করার পর academyName ভ্যারিয়েবলের মান হয় অসংজ্ঞায়িত(undefined)

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>জাভাস্ক্রিপ্ট পাত্রসমূহ</h3>

<h4> যদি একটি পাত্র কোনো মান ছাড়া ঘোষণা করা হয় তবে এর ভ্যালু অনির্দিষ্ট দেখাবে।</h4>

<p id="test"></p>

<script>

var academyName;

document.getElementById("test").innerHTML = academyName;

</script>

</body>

</html>

Copy

**ফলাফল**

### জাভাস্ক্রিপ্ট পাত্রসমূহ

#### যদি একটি পাত্র কোনো মান ছাড়া ঘোষণা করা হয় তবে এর ভ্যালু অনির্দিষ্ট দেখাবে।

undefined

**জাভাস্ক্রিপ্ট ভ্যারিয়েবলকে পুনরায় ঘোষণা**

জাভাস্ক্রিপ্ট ভ্যারিয়েবলকে পুনরায় ঘোষণা করলেও আগের মান হারায় না।

স্টেটমেন্টটি সম্পাদন হওয়ার পরেও acadmeyName ভ্যারিয়েবলের মান "স্যাট" থাকেঃ

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>জাভাস্ক্রিপ্ট পাত্রসমূহ</h3>

<h4>যদি আপনি জাভাস্ক্রিপ্টের পাত্রের মান পুনরায় ঘোষণা করেন তবে এটা তার মান হারাবে না।</h4>

<p id="test"></p>

<script>

var academyName = "Satt";

var academyName;

document.getElementById("test").innerHTML = academyName ;

</script>

</body>

</html>

Copy

**ফলাফল**

### জাভাস্ক্রিপ্ট পাত্রসমূহ

#### যদি আপনি জাভাস্ক্রিপ্টের পাত্রের মান পুনরায় ঘোষণা করেন তবে এটা তার মান হারাবে না।

Satt

**জাভাস্ক্রিপ্ট গণিত**

বীজগণিতের মতই জাভাস্ক্রিপ্টের ভ্যারিয়েবলগুলোকে অপারেটরের সাহায্যে গাণিতিক যোগ-বিয়োগ করা যায়ঃ

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>জাভাস্ক্রিপ্ট পাত্রসমূহ</h3>

<h6>10,23 ও 13 যোগ করে এর ফলাফল প্রদর্শন করান।</h6>

<p id="test"></p>

<script>

var a = 10 + 23 + 13;

document.getElementById("test").innerHTML = a;

</script>

</body>

</html>

Copy

**ফলাফল**

### জাভাস্ক্রিপ্ট পাত্রসমূহ

###### 10,23 ও 13 যোগ করে এর ফলাফল প্রদর্শন করান।

46

আপনি স্ট্রিংকেও যোগ করতে পারেন। কিন্তু এগুলো যোগ করলে যোগের পরিবর্তে পাশাপাশি(concatination) বসেঃ

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>জাভাস্ক্রিপ্ট পাত্রসমূহ</h3>

<h6>যোগ করুন "আজিজুর " + " " + "রহমান":</h6>

<p id="test"></p>

<script>

var a = "আজিজুর " + " " + "রহমান";

document.getElementById("test").innerHTML = a;

</script>

</body>

</html>

Copy

**ফলাফল**

### জাভাস্ক্রিপ্ট পাত্রসমূহ

###### যোগ করুন "আজিজুর " + " " + "রহমান":

আজিজুর রহমান

নিচেরটাও চেষ্টা করি

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>জাভাস্ক্রিপ্ট ভেরিয়েবল সমুহ </h3>

<h6>যোগ করুন "7" + 4 + 1 এবং এর ফলাফল প্রদর্শন করান।</h6>

<p id="test"></p>

<script>

var a = "7" + 4 + 1;

document.getElementById("test").innerHTML = a;

</script>

</body>

</html>

Copy

**ফলাফল**

### জাভাস্ক্রিপ্ট ভেরিয়েবল সমুহ

###### যোগ করুন "7" + 4 + 1 এবং এর ফলাফল প্রদর্শন করান।

741

|  |  |
| --- | --- |
|  | যদি আপনি একটি সংখ্যাকে কোটেশনের মধ্যে রাখেন তাহলে পরবর্তী সংখ্যাগুলোও স্ট্রিং-এ রূপান্তরিত হয় এবং এগুলো পাশাপাশি বসে। |

## জাভাস্ক্রিপ্ট স্টেটমেন্ট

নিচের স্টেটমেন্টটি ব্রাউজারকে id="test" সম্বলিত এইচটিএমএল এলিমেন্টের মধ্যে "Satt Academy" লিখতে বলেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h5>এইচটিএমএল এর মধ্যে জাভাস্ক্রিপ্ট স্টেটমেন্ট ব্রাউজারকে নির্দেশ করে।</h5>

<p id="test"></p>

<script>

document.getElementById("test").innerHTML = "Satt Academy";

</script>

</body>

</html>

Copy

### ফলাফল

##### এইচটিএমএল এর মধ্যে জাভাস্ক্রিপ্ট স্টেটমেন্ট ব্রাউজারকে নির্দেশ করে।

Satt Academy

## জাভাস্ক্রিপ্ট প্রোগ্রাম

অধিকাংশ জাভাস্ক্রিপ্ট প্রোগ্রামেই অনেক জাভাস্ক্রিপ্ট স্টেটমেন্ট থাকে।

স্টেটমেন্টগুলো ক্রমানুসারে একের পর এক সম্পাদিত হয়।

নিচের উদাহরণে প্রথমে x, y এবং z এর মান নির্ধারণ করা হয়। পরিশেষে z এর মান প্রদর্শিত হয়।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h5>জাভাস্ক্রিপ্ট স্টেটমেন্টের ব্লক হলো জাভাস্ক্রিপ্ট কোড</h5>

<p id="test"></p>

<script>

var a = 12;

var b = 13;

var c = a + b;

document.getElementById("test").innerHTML = c;

</script>

</body>

</html>

Copy

### ফলাফল

##### জাভাস্ক্রিপ্ট স্টেটমেন্টের ব্লক হলো জাভাস্ক্রিপ্ট কোড

25

|  |  |
| --- | --- |
|  | জাভাস্ক্রিপ্ট প্রোগ্রাম এবং জাভাস্ক্রিপ্ট স্টেটমেন্টকে জাভাস্ক্রিপ্ট কোড বলা হয়। |

## সেমিকোলন (;)

সেমিকোলনের মাধ্যমে জাভাস্ক্রিপ্ট স্টেটমেন্টগুলোকে আলাদা করা হয়।

সম্পাদনযোগ্য প্রত্যেক স্টেটমেন্টের পরে সেমিকোলন যোগ করুনঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>জাভাক্রিপ্ট স্টেটমেন্টগুলো সেমিকোলন দ্বারা আলাদা করা হয়।</h4>

<p id="test"></p>

<script>

a = 1;

b = 2;

c = a + b;

document.getElementById("test").innerHTML = c;

</script>

</body>

</html>

Copy

### ফলাফল

#### জাভাক্রিপ্ট স্টেটমেন্টগুলো সেমিকোলন দ্বারা আলাদা করা হয়।

3

সেমিকোলনের মাধ্যমে স্টেটমেন্টকে আলাদা করলে একই লাইনে অনেক স্টেটমেন্ট লেখা যায়ঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>জাভাক্রিপ্ট স্টেটমেন্টগুলো সেমিকোলন দ্বারা আলাদা করা হয়।</h4>

<p id="test"></p>

<script>

a = 1; b = 2; c = a + b;

document.getElementById("test").innerHTML = c;

</script>

</body>

</html>

Copy

### ফলাফল

#### জাভাক্রিপ্ট স্টেটমেন্টগুলো সেমিকোলন দ্বারা আলাদা করা হয়।

3

|  |  |
| --- | --- |
|  | অনেকক্ষেত্রে আপনি সেমিকোলন ছাড়াও স্টেটমেন্ট দেখতে পারেন।  সেমিকোলনের মাধ্যমে স্টেটমেন্টকে আলাদা না করলেও চলে, কিন্তু আমরা আপনাকে সেমিকোলন ব্যবহার করতে সুপারিশ করছি। |

## জাভাস্ক্রিপ্ট স্পেস

জাভাস্ক্রিপ্ট একের অধিক স্পেসকে এড়িয়ে চলে। শুধুমাত্র আপনার পড়ার সুবিধার্থে অতিরিক্ত স্পেস যোগ করতে পারেন।

নিচের লাইনগুলো একই রকমঃ

var person = "আজিজ";

var person="আজিজ";

Copy

অপারেটরের ( = + - \* / ) উভয়পাশে স্পেস রাখা একটি ভাল অভ্যাসঃ

var a = b + c;

Copy

## জাভাস্ক্রিপ্ট লাইনের দৈর্ঘ্য এবং লাইনের বিরতি

অধিক স্পষ্টতার জন্য প্রোগ্রামাররা ৮০ অক্ষরের চাইতে বড় লাইনের কোডকে এড়িয়ে চলে

যদি জাভাস্ক্রিপ্ট স্টেটমেন্ট এক লাইনে সম্পূর্ণ করা না যায়, তাহলে অপারেটরের পরে লাইন ব্রেক করে পরবর্তী লাইনে বাকী অংশটুকু লিখুনঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h5> একটি অপারেটর অথবা একটি কমার পরে কোডের লাইন ব্রেক করা যেতে পারে।

</h5>

<p id="test"></p>

<script>

document.getElementById("test").innerHTML =

"স্যাট সুহিন";

</script>

</body>

</html>

Copy

### ফলাফল

##### একটি অপারেটর অথবা একটি কমার পরে কোডের লাইন ব্রেক করা যেতে পারে।

স্যাট সুহিন

## জাভাস্ক্রিপ্ট কোড ব্লক

জাভাস্ক্রিপ্ট স্টেটমেন্টগুলোকে দ্বিতীয় বন্ধনীর{...} মাধ্যমে কোড ব্লকের মাধ্যমে একত্রিত করা যায়।

কোড ব্লকের উদ্দেশ্য হচ্ছে স্টেটমেন্টগুলোকে একসাথে সম্পাদন করা।

জাভাস্ক্রিপ্টে মাঝে মাঝে স্টেটমেন্টগুলোকে ব্লকের মধ্যে একসাথে দেখতে পাবেন সেটা হচ্ছে জাভাস্ক্রিপ্ট ফাংশনঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>আমার ওয়েব পেইজ</h3>

<h5 id="test">এটি একটি অনুচ্ছেদ</h5>

<p id="para">এটি একটি প্যারাগ্রাফ</p>

<p>

<button type="button" onclick="myFunc()">ক্লিক করি</button>

</p>

<script>

function myFunc() {

document.getElementById("test").innerHTML = "হ্যালো শাপলা";

document.getElementById("para").innerHTML = "তুমি কেমন আছো?";

}

</script>

<p>যখন আপনি বাটনে ক্লিক করবেন তখন দুইটি উপাদান পরিবর্তন হবে।</p>

</body>

</html>

Copy

### ফলাফল

### আমার ওয়েব পেইজ

##### এটি একটি অনুচ্ছেদ

এটি একটি প্যারাগ্রাফ

যখন আপনি বাটনে ক্লিক করবেন তখন দুইটি উপাদান পরিবর্তন হবে।

|  |  |
| --- | --- |
|  | পরবর্তী টিউটোরিয়ালে ফাংশন সম্মন্ধে আপনি আরো শিখবেন। |

## জাভাস্ক্রিপ্ট কিওয়ার্ড

জাভাস্ক্রিপ্টে নির্দিষ্ট কার্য সম্পাদন করার জন্য প্রায়ই কিওয়ার্ড দিয়ে স্টেটমেন্ট শুরু হয়।

এখানে কিছু কিওয়ার্ডের তালিকা দেওয়া আছে যা আপনি এই টিউটোরিয়ালে শিখবেনঃ

|  |  |
| --- | --- |
| **কিওয়ার্ড** | **বর্ণনা** |
| break | switch অথবা loop কে বন্ধ করে দেয় |
| continue | লুপ হতে বের হয়ে যায় এবং উপর হতে পূনরায় শুরু করে |
| debugger | জাভাস্ক্রিপ্ট এক্সিকিউশনকে থামিয়ে দেয় এবং ডিবাগিং ফাংশন থাকলে তাকে কল করে |
| do ... while | প্রথমে do ব্লকের স্টেটমেন্ট সম্পাদিত হয় এবং while এর শর্ত সত্য হলে ব্লকটির পুনরাবৃত্তি ঘটে |
| for | যতক্ষণ শর্ত সত্য থাকবে ততক্ষণ একটি ব্লকের স্টেটমেন্টগুলো সম্পাদন হতে থাকবে |
| function | ফাংশন ঘোষণা করে |
| if ... else | শর্তের উপর ভিত্তি করে যেকোন একটি স্টেটমেন্টের ব্লক সম্পাদিত হয় |
| return | ফাংশন থেকে বের হয়ে যায় |
| switch | বিভিন্ন অবস্থা/ঘটনার উপর ভিত্তি করে একটি স্টেটমেন্টের ব্লক একবার সম্পাদিত হয় |
| try ... catch | একটি স্টেটমেন্টের ব্লকে ভুল নিয়ন্ত্রণ করে |
| var | চলক বা ভ্যারিয়েবল ডিক্লেয়ার করে |

|  |  |
| --- | --- |
|  | জাভাস্ক্রিপ্ট কিওয়ার্ডগুলো সংরক্ষিত শব্দ যেগুলো ভ্যারিয়েবলের নামের জন্য ব্যবহার করা যাবেনা। |
|  | জাভাস্ক্রিপ্ট কমেন্ট [« পূরবর্তী](https://www.sattacademy.com/js/js_statements.php)  [পরবর্তী »](https://www.sattacademy.com/js/js_operators.php)  জাভাস্ক্রিপ্ট কোডকে ব্যাখ্যা করার জন্য এবং অধিক পাঠযোগ্য করে তোলার জন্য জাভাস্ক্রিপ্ট কমেন্ট ব্যবহার করা হয়।  কোডকে সম্পাদন থেকে বিরত রাখার জন্যও জাভাস্ক্রিপ্ট কমেন্ট ব্যবহার করা হয়। বিশেষ করে বিকল্প কোনো কোডকে টেস্ট করার সময় কমেন্ট ব্যবহার করা হয়। এক লাইনের কমেন্ট এক লাইনের কমেন্ট // দিয়ে শুরু হয়।  // এর পরে যেকোন ধরণের টেক্সট কে জাভাস্ক্রিপ্ট এড়িয়ে চলে।  নিচের উদাহরণে কোডকে ব্যাখ্যা করার জন্য প্রতি লাইনের আগে এক লাইনের কমেন্ট করা হয়েছে। উদাহরণ <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>জাভাস্ক্রিপ্ট উদাহরণ</title>  </head>  <body>  <h3 id="header"></h3>  <p id="para"></p>  <script>  // হেডিং পরিবর্তন হবে  document.getElementById("header").innerHTML = "প্রথম পেজ";  // প্যারাগ্রাফ পরিবর্তন হবে  document.getElementById("para").innerHTML = "প্রথম অনুচ্ছেদ";  </script>  <p><strong>নোটঃ</strong> কমেন্টগুলো সম্পাদন হবে না।</p>  </body>  </html>  Copy ফলাফলপ্রথম পেজ প্রথম অনুচ্ছেদ  নোটঃ কমেন্টগুলো সম্পাদন হবে না।  নিচের উদাহরণে কোডের বর্ণনা করার জন্য প্রতি লাইন কোডের শেষে এক লাইনের কমেন্ট যোগ করা হয়েছেঃ উদাহরণ <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>জাভাস্ক্রিপ্ট উদাহরণ</title>  </head>  <body>  <p id="test"></p>  <script>  var a = 10; // a কে চলক ঘোষণা করে এর মান 10 দেওয়া হয়েছে  var b = a + 5; // b কে চলক ঘোষণা করে এর মান a + 5 দেওয়া হয়েছে  document.getElementById("test").innerHTML = b; // b এর মান test তে লেখা হলো  </script>  <p><strong>নোটঃ</strong> কমেন্টগুলো সম্পাদন হবে না।</p>  </bodb>  </html>  Copy ফলাফল 15  **নোটঃ** কমেন্টগুলো সম্পাদন হবে না। একের অধিক লাইনের কমেন্ট একের অধিক লাইনের কমেন্ট /\* দিয়ে শুরু হয় এবং \*/ দিয়ে শেষ হয়।  /\* এবং \*/ এর মধ্যে যেকোন ধরণের টেক্সটকে জাভাস্ক্রিপ্ট এড়িয়ে চলে।  নিচের উদাহরণে কোড বুঝার জন্য একের অধিক লাইনের কমেন্ট ব্যবহার করা হয়েছে: উদাহরণ <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>জাভাস্ক্রিপ্ট উদাহরণ</title>  </head>  <body>  <h3 id="header"></h3>  <p id="para"></p>  <script>  /\*  এই কোডগুলো এইচটিএমএল ওয়েব পেজের  id = "header" এর হেডিং এবং  id = "para" এর প্যারাগ্রাফ পরিবর্তন করে  \*/  document.getElementById("header").innerHTML = "প্রথম পেজ";  document.getElementById("para").innerHTML = "প্রথম অনুচ্ছেদ";  </script>  <p><strong>নোটঃ</strong> কমেন্টগুলো সম্পাদন হবে না।</p>  </body>  </html>  Copy ফলাফলপ্রথম পেজ প্রথম অনুচ্ছেদ  নোটঃ কমেন্টগুলো সম্পাদন হবে না।   |  |  | | --- | --- | |  | অধিকাংশক্ষেত্রেই এক লাইনের কমেন্ট ব্যবহার করা হয়। ডকুমেন্ট লেখার ক্ষেত্রে একের অধিক লাইনের কমেন্ট লেখা হয়। |  কমেন্টের মাধ্যমে কোড সম্পাদনে বাধা দেওয়া কোড টেস্টের ক্ষেত্রে কমেন্টের মাধ্যমে কোড সম্পাদনকে বাধা দেওয়া উচিত।  নিচের উদাহরণে // ব্যবহার করে প্রথম লাইনকে সম্পাদন থেকে বাধা দেওয়া হয়েছেঃ উদাহরণ <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>জাভাস্ক্রিপ্ট উদাহরণ</title>  </head>  <body>  <h3 id="header"></h3>  <p id="para"></p>  <script>  //document.getElementById("header").innerHTML = "প্রথম পেজ";  document.getElementById("para").innerHTML = "প্রথম অনুচ্ছেদ";  </script>  <p><strong>নোটঃ</strong> কমেন্টগুলো সম্পাদন হবে না।</p>  </body>  </html>  Copy ফলাফল প্রথম অনুচ্ছেদ  **নোটঃ** কমেন্টগুলো সম্পাদন হবে না।  নিচের উদাহরণে একাধিক লাইনের কোডকে সম্পাদন থেকে বাধা দিতে কমেন্ট ব্লক ব্যবহার করা হয়েছেঃ উদাহরণ <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>জাভাস্ক্রিপ্ট উদাহরণ</title>  </head>  <body>  <h3 id="header"></h3>  <p id="para"></p>  <script>  /\*  document.getElementById("header").innerHTML = "প্রথম পেজ";  document.getElementById("para").innerHTML = "প্রথম অনুচ্ছেদ";  \*/  </script>  <p><strong>নোটঃ</strong> কমেন্টগুলো সম্পাদন হবে না।</p>  </body>  </html>  Copy ফলাফল নোটঃ কমেন্টগুলো সম্পাদন হবে না। |
|  | জাভাস্ক্রিপ্ট অপারেটর(Operator)উদাহরণ <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>জাভাস্ক্রিপ্ট উদাহরণ</title>  </head>  <body>  <h4>জাভাস্ক্রিপ্ট অপারেটর</h4>  <h5>a = 10, b = 5,হিসাব করুন c = a + b, এবং c এর মান প্রদর্শন করান।</h5>  <p id="test"></p>  <script>  var a = 10;  var b = 5;  var c = a + b;  document.getElementById("test").innerHTML = c;  </script>  </body>  </html>  Copy ফলাফলজাভাস্ক্রিপ্ট অপারেটরa = 10, b = 5,হিসাব করুন c = a + b, এবং c এর মান প্রদর্শন করান। 15 জাভাস্ক্রিপ্ট গাণিতিক অপারেটর সংখ্যার যোগ-বিয়োগ করার জন্য জাভাস্ক্রিপ্টে গাণিতিক অপারেটর ব্যবহার করা হয়ঃ   |  |  | | --- | --- | | অপারেটর | বর্ণনা | | + | যোগ(Addition) | | - | বিয়োগ(Subtraction) | | \* | গুণ(Multiplication) | | / | ভাগ(Division) | | % | (ভাগশেষ)Modulus | | ++ | এক করে বৃদ্ধি (Increment) | | -- | এক করে হ্রাস(Decrement) |   যোগ(+) অপারেটর সংখ্যাকে যোগ করেঃ যোগ <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>জাভাস্ক্রিপ্ট উদাহরণ</title>  </head>  <body>  <h4>+ অপারেটর</h4>  <p id="test"></p>  <script>  var a = 25;  var b = 40;  var c = 13;  var d = a + b + c;  document.getElementById("test").innerHTML = d;  </script>  </body>  </html>  Copy ফলাফল+ অপারেটর 78  গুণন(\*) অপারেটর সংখ্যাকে গুণ করেঃ গুণ <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>জাভাস্ক্রিপ্ট উদাহরণ</title>  </head>  <body>  <h4>\* অপারেটর</h4>  <p id="test"></p>  <script>  var a = 40;  var b = 13;  var c = a \* b;  document.getElementById("test").innerHTML = c;  </script>  </body>  </html>  Copy ফলাফল\* অপারেটর 520   |  |  | | --- | --- | |  | পরবর্তী অধ্যায়ে জাভাস্ক্রিপ্ট অপারেটর সম্পর্কে আরো অনেক শিখবেন। |  জাভাস্ক্রিপ্ট এসাইনমেন্ট অপারেটর এসাইনমেন্ট অপারেটর জাভাস্ক্রিপ্ট ভ্যারিয়েবলের মধ্যে ভ্যালু এসাইন করে।   |  |  |  | | --- | --- | --- | | অপারেটর | উদাহরণ |  | | = | x = y | x = y | | += | x += y | x = x + y | | -= | x -= y | x = x - y | | \*= | x \*= y | x = x \* y | | /= | x /= y | x = x / y | | %= | x %= y | x = x % y |   এসাইনমেন্ট অপারেটর(=) জাভাস্ক্রিপ্ট ভ্যারিয়েবলের মধ্যে ভ্যালু জমা রাখে। উদাহরণ <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>জাভাস্ক্রিপ্ট উদাহরণ</title>  </head>  <body>  <h3>= অপারেটর</h3>  <p id="test"></p>  <script>  var a = 32;  document.getElementById("test").innerHTML = a;  </script>  </body>  </html>  Copy ফলাফল= অপারেটর 32  যোগ এসাইনমেন্ট(+=) অপারেটর ভ্যারিয়েবলে ভ্যালু যোগ করে। উদাহরণ <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>জাভাস্ক্রিপ্ট উদাহরণ</title>  </head>  <body>  <h3>+= অপারেটর</h3>  <p id="test"></p>  <script>  var a = 32;  a += 8;  document.getElementById("test").innerHTML = a;  </script>  </body>  </html>  Copy ফলাফল+= অপারেটর 40 জাভাস্ক্রিপ্ট স্ট্রিং অপারেটর স্ট্রিংকে যোগ(concaenate) করতেও + অপারেটর ব্যবহার করতে হয়।   |  |  | | --- | --- | |  | যখন যোগ(+) অপারেটর স্ট্রিং-এ ব্যবহার করা হয় তখন একে কনকাটিনেশন(concatenation) অপারেটর বলা হয়। |  উদাহরণ <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>জাভাস্ক্রিপ্ট উদাহরণ</title>  </head>  <body>  <h4>জাভাস্ক্রিপ্ট অপারেটর</h4>  <h5> + অপারেটর স্ট্রিংগুলোকে সংযুক্ত করে</h5>  <p id="test"></p>  <script>  var txt1 = "আজিজুর";  var txt2 = "রহমান ";  document.getElementById("test").innerHTML = txt1 + " " + txt2;  </script>  </body>  </html>  Copy ফলাফলজাভাস্ক্রিপ্ট অপারেটর+ অপারেটর স্ট্রিংগুলোকে সংযুক্ত করে আজিজুর রহমান  += এসাইনমেন্ট অপারেটর স্ট্রিংকে যোগ করার ক্ষেত্রে ব্যবহার করা হয়ঃ উদাহরণ <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>জাভাস্ক্রিপ্ট উদাহরণ</title>  </head>  <body>  <h4>জাভাস্ক্রিপ্ট অপারেটর</h4>  <p>এসাইনমেন্ট অপারেটর += দিয়ে স্ট্রিং গুলোকে সংযুক্ত করতে পারে।</p>  <p id="test"></p>  <script>  var txt1 = "আজিজুর";  txt1 += " রহমান ";  document.getElementById("test").innerHTML = txt1;  </script>  </body>  </html>  Copy ফলাফলজাভাস্ক্রিপ্ট অপারেটর এসাইনমেন্ট অপারেটর += দিয়ে স্ট্রিং গুলোকে সংযুক্ত করতে পারে।  আজিজুর রহমান স্ট্রিং এবং সংখ্যার যোগ দুটি সংখ্যাকে যোগ করলে যোগফল সংখ্যা হবে কিন্তু একটি সংখ্যা ও একটি স্ট্রিং যোগ করলে যোগফল হবে স্ট্রিংঃ উদাহরণ <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>জাভাস্ক্রিপ্ট উদাহরণ</title>  </head>  <body>  <h4>জাভাস্ক্রিপ্ট অপারেটর</h4>  <h5>একটি নাম্বার এবং একটি স্ট্রিং যুক্ত করে স্ট্রিং রিটার্ন করে ।</h5>  <p id="test"></p>  <script>  var a = 23 + 23;  var b = "23" + 23;  var c = "operator" + 23;  document.getElementById("test").innerHTML =  a + "<br>" + b + "<br>" + c;  </script>  </body>  </html>  Copy ফলাফলজাভাস্ক্রিপ্ট অপারেটরএকটি নাম্বার এবং একটি স্ট্রিং যুক্ত করে স্ট্রিং রিটার্ন করে । 46 2323 operator23  নিয়মঃ **নাম্বার এবং স্ট্রিংকে যোগ করলে যোগফল হবে স্ট্রিং!** জাভাস্ক্রিপ্ট কম্প্যারিজন(Comparison) এবং লজিক্যাল(Logical) অপারেটর  |  |  | | --- | --- | | অপারেটর | বর্ণনা | | == | সমান | | === | সমান মান এবং একই টাইপ | | != | সমান না | | !== | একই মান এবং টাইপের না | | > | বড় | | < | ছোট | | >= | বড় অথবা সমান | | <= | ছোট অথবা সমান | | ? | তিনটি অপারেটর নিয়ে গঠিত (if...else এর মতই কাজ করে) | |  | জেএস কম্প্যারিজন অধ্যায়ে কম্প্যারিজন এবং লজিক্যাল অপারেটর সম্পর্কে সম্পূর্ণ বর্ণনা করা হয়েছে। |  জাভাস্ক্রিপ্ট টাইপ অপারেটর  |  |  | | --- | --- | | অপারেটর | বর্ণনা | | typeof | ভ্যারিয়েবলের টাইপ রিটার্ন করে | | instanceof | যদি একটি অবজেক্ট অন্য একটি অবজেক্ট টাইপের হয় তাহলে true রিটার্ন করবে | |  | জেএস ধরন পরিবর্তন অধ্যায়ে টাইপ অপারেটর সম্পর্কে আলোচনা করা হয়েছে। | |

## জাভাস্ক্রিপ্ট গাণিতিক অপারেটর(Arithmetic Operator)

সংখ্যা(লিটারাল অথবা ভ্যারিয়েবল) যোগ-বিয়োগ করার জন্য জাভাস্ক্রিপ্টে গাণিতিক অপারেটর ব্যবহার করা হয়।

|  |  |
| --- | --- |
| অপারেটর | বর্ণনা |
| + | যোগ(Addition) |
| - | বিয়োগ(Subtraction) |
| \* | গুণ(Multiplication) |
| / | ভাগ(Division) |
| % | মডুলাস(Modulus) |
| ++ | এক করে বৃদ্ধি (Increment) |
| -- | এক করে হ্রাস(Decrement) |

## গাণিতিক অপারেটর(Arithmetic Operation)

গাণিতিক অপারেটর সাধারণত দুইটি নাম্বারের হিসাব-নিকাশ করে।

সংখ্যা দুইটি লিটারাল হতে পারেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>একটি প্রতীকস্বরূপ গানিতিক ক্রিয়া দুইটি সংখ্যা নেয় এবং একটি নতুন সংখ্যা প্রকাশ করে।</h4>

<p id="test"></p>

<script>

var a = 25 + 40 + 13;

document.getElementById("test").innerHTML = a;

</script>

</body>

</html>

Copy

### ফলাফল

#### একটি প্রতীকস্বরূপ গানিতিক ক্রিয়া দুইটি সংখ্যা নেয় এবং একটি নতুন সংখ্যা প্রকাশ করে।

78

অথবা ভ্যারিয়েবল হতে পারেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>একটি প্রতীকস্বরূপ গানিতিক ক্রিয়া দুইটি সংখ্যা নেয় এবং একটি নতুন সংখ্যা প্রকাশ করে।</h4>

<p id="test"></p>

<script>

var a = 25;

var b = 40;

var c = 13;

var d = a + b + c;

document.getElementById("test").innerHTML = d;

</script>

</body>

</html>

Copy

### ফলাফল

#### একটি প্রতীকস্বরূপ গানিতিক ক্রিয়া দুইটি সংখ্যা নেয় এবং একটি নতুন সংখ্যা প্রকাশ করে।

78

অথবা এক্সপ্রেশন(expression) হতে পারেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>একটি প্রতীকস্বরূপ গানিতিক ক্রিয়া দুইটি সংখ্যা নেয় এবং একটি নতুন সংখ্যা প্রকাশ করে।</h4>

<p id="test"></p>

<script>

var a = 25;

var b = (12 - 2) \* a;

document.getElementById("test").innerHTML = b;

</script>

</body>

</html>

Copy

### ফলাফল

#### একটি প্রতীকস্বরূপ গানিতিক ক্রিয়া দুইটি সংখ্যা নেয় এবং একটি নতুন সংখ্যা প্রকাশ করে।

250

## অপারেটর এবং অপারেন্ড(Operator and Operand)

গাণিতিক হিসাব-নিকাশে সংখ্যাকে বলা হয় অপারেন্ড।

অপারেটর(operator) দুইটি অপারেন্ডের কার্য নির্ধারন করে।

|  |  |  |
| --- | --- | --- |
| অপারেন্ড | অপারেটর | অপারেন্ড |
| 100 | + | 50 |

যোগ(+) অপারেটর সংখ্যার যোগ করেঃ

### যোগ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>+ অপারেটর</h4>

<p id="test"></p>

<script>

var a = 25;

var b = 40;

var c = 13;

var d = a + b + c;

document.getElementById("test").innerHTML = d;

</script>

</body>

</html>

Copy

### ফলাফল

#### + অপারেটর

78

বিয়োগ(-) অপারেটর সংখ্যার বিয়োগ করেঃ

### বিয়োগ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>- অপারেটর</h4>

<p id="test"></p>

<script>

var a = 40;

var b = 13;

var c = a - b;

document.getElementById("test").innerHTML = c;

</script>

</body>

</html>

Copy

### ফলাফল

#### - অপারেটর

27

গুণ(\*) অপারেটর সংখ্যার গুণ করে

### গুণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>\* অপারেটর</h4>

<p id="test"></p>

<script>

var a = 40;

var b = 13;

var c = a \* b;

document.getElementById("test").innerHTML = c;

</script>

</body>

</html>

Copy

### ফলাফল

#### \* অপারেটর

520

ভাগ(/) অপারেটর সংখ্যার ভাগ করেঃ

### ভাগ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>/ অপারেটর</h4>

<p id="test"></p>

<script>

var a = 40;

var b = 10;

var c = a / b;

document.getElementById("test").innerHTML = c;

</script>

</body>

</html>

Copy

### ফলাফল

#### / অপারেটর

4

ভাগশেষ(%) অপারেটরের মাধ্যমে ভাগশেষ পাওয়া যায়।

### ভাগশেষ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>% অপারেটর</h4>

<p id="test"></p>

<script>

var a = 40;

var b = 13;

var c = a % b;

document.getElementById("test").innerHTML = c;

</script>

</body>

</html>

Copy

### ফলাফল

#### % অপারেটর

1

বৃদ্ধি(++) অপারেটর সংখ্যাকে এক করে বৃদ্ধি করেঃ

### বৃদ্ধি

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>++ অপারেটর</h4>

<p id="test"></p>

<script>

var a = 40;

a++;

var b = a;

document.getElementById("test").innerHTML = b;

</script>

</body>

</html>

Copy

### ফলাফল

#### ++ অপারেটর

41

হ্রাস(--) অপারেটর সংখ্যার এক করে হ্রাস করেঃ

### হ্রাস

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>-- অপারেটর</h4>

<p id="test"></p>

<script>

var a = 40;

a--;

var b = a;

document.getElementById("test").innerHTML = b;

</script>

</body>

</html>

Copy

### ফলাফল

#### -- অপারেটর

39

## অপারেটর অগ্রাধিকার

গাণিতিক এক্সপ্রেশনে বিভিন্ন অপারেটর(+, -, \*, / % ) তাদের ক্রমানুসারে সংখ্যাকে অপারেট করে।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>যোগ করার আগে গুন করার উপর জোর দেওয়া হয়।</h4>

<p id="test"></p>

<script>

document.getElementById("test").innerHTML = 20 + 5 \* 5;

</script>

</body>

</html>

Copy

### ফলাফল

#### যোগ করার আগে গুন করার উপর জোর দেওয়া হয়।

45

উপরের উদাহরণের ফলাফল কি 25 \* 5 এর মত হবে নাকি 20 + 25 এর মত হবে?

যোগ এবং গুণের মধ্যে কোনটি আগে সংঘটিত হবে?

আমরা স্কুল যেমন গণিতে গুণের কাজ আগে করতাম, এখানেও তেমনি গুণের কাজ আগে হয়েছে।

যোগ(+) এবং বিয়োগ(-) এর তুলনায় গুণ(\*) এবং ভাগের(/) অগ্রাধিকার বেশি।

স্কুলের মত এখানেও বন্ধনী(parentheses) ব্যবহার করে অপারেটরের অগ্রাধিকার পরিবর্তন করা যায়।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>যোগ করার আগে গুন করার উপর জোর দেওয়া হয়।</h4>

<h5>কিন্তু গুন করার আগে প্রথম বন্ধনীর কাজ করা হয়।</h5>

<p id="test"></p>

<script>

document.getElementById("test").innerHTML = (14 + 16) \* 3;

</script>

</body>

</html>

Copy

### ফলাফল

#### যোগ করার আগে গুন করার উপর জোর দেওয়া হয়।

##### কিন্তু গুন করার আগে প্রথম বন্ধনীর কাজ করা হয়।

90

বন্ধনী ব্যবহার করলে সবার আগে বন্ধনীর কাজ হয়

যখন অনেক অপারেশনের একই অগ্রাধিকার থাকে(যেমন - যোগ এবং বিয়োগ) তখন বাম দিক থেকে অপারেশন শুরু হয়ে ডান দিকে শেষ হয়ঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>যখন অনেকগুলো সাধারন কাজ থাকে তখন তা বাম থেকে ডানের দিকে হিসেব করা হয়।</h4>

<p id="test"></p>

<script>

document.getElementById("test").innerHTML = 20 + 5 - 5;

</script>

</body>

</html>

Copy

### ফলাফল

#### যখন অনেকগুলো সাধারন কাজ থাকে তখন তা বাম থেকে ডানের দিকে হিসেব করা হয়।

20

## জাভাস্ক্রিপ্ট অপারেটর অগ্রাধিকারের মান

|  |  |  |  |
| --- | --- | --- | --- |
| মান | অপারেটর | বর্ণনা | উদাহরণ |
| ১৯ | ( ) | এক্সপ্রেশন শ্রেণীবদ্ধকরণ | (5 + 6) |
|  |  |  |  |
| ১৮ | . | মেম্বার | person.name |
| ১৮ | [] | মেম্বার | person["name"] |
|  |  |  |  |
| ১৭ | () | ফাংশন কল | myFunction() |
| ১৭ | new | তৈরি করে | new Date() |
|  |  |  |  |
| ১৬ | ++ | পরে বৃ্দ্ধি করে | i++ |
| 16 | -- | পরে হ্রাস করে | i-- |
|  |  |  |  |
| 15 | ++ | আগে বৃ্দ্ধি করে | ++i |
| 15 | -- | আগে হ্রাস করে | --i |
| 15 | ! | লজিক্যাল না বুঝায় | !(x==y) |
| ১৫ | typeof | টাইপ | typeof x |
|  |  |  |  |
| ১৪ | \* | গুণ | ১০ \* ৫ |
| ১৪ | / | ভাগ | ১০ / ৫ |
| 14 | % | ভাগশেষ | ১০ % ৫ |
| 14 | \*\* | বর্গ করে | 10 \*\* 2 |
|  |  |  |  |
| ১৩ | + | যোগ | ১০ + ৫ |
| ১৩ | - | বিয়োগ | ১০ - ৫ |
|  |  |  |  |
| ১২ | << | বাম দিকে নেয়(Shift left) | x << 2 |
| ১২ | >> | ডান দিকে নেয়(Shift right) | x >> 2 |
|  |  |  |  |
| ১১ | < | ছোট | x < y |
| ১১ | <= | ছোট অথবা সমান | x <= y |
| 11 | > | বড় | x > y |
| 11 | >= | বড় অথবা সমান | x >= y |
|  |  |  |  |
| ১০ | == | সমান | x == y |
| ১০ | === | একই মান এবং টাইপ | x === y |
| ১০ | != | সমান না | x != y |
| ১০ | !== | একই মান এবং টাইপ নয় | x !== y |
|  |  |  |  |
| ৬ | && | এবং | x && y |
| ৫ | || | অথবা | x || y |
|  |  |  |  |
| ৩ | = | ভ্যালু জমা রাখা | x = y |
| ৩ | += | ভ্যালু জমা রাখা | x += y |
| ৩ | -= | ভ্যালু জমা রাখা | x -= y |
| ৩ | \*= | ভ্যালু জমা রাখা | x \*= y |
| ৩ | /= | ভ্যালু জমা রাখা | x /= y |

|  |  |
| --- | --- |
|  | লাল চিহ্নিত ঘরের অপারেটরটি পরীক্ষামূলক বুঝাতে ব্যবহার করা হয়েছে। |

## জাভাস্ক্রিপ্ট এসাইনমেন্ট অপারেটর

এসাইনমেন্ট অপারেটর জাভাস্ক্রিপ্ট ভ্যারিয়েবলে ভ্যালু জমা রাখে।

|  |  |  |
| --- | --- | --- |
| অপারেটর | উদাহরণ | উদাহরণ কলামের মত |
| = | x = y | x = y |
| += | x += y | x = x + y |
| -= | x -= y | x = x - y |
| \*= | x \*= y | x = x \* y |
| /= | x /= y | x = x / y |
| %= | x %= y | x = x % y |

"=" এসাইনমেন্ট অপারেটরটি ভ্যারিয়েবলের মধ্যে মান জমা রাখে।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>= অপারেটর</h3>

<p id="test"></p>

<script>

var a = 32;

document.getElementById("test").innerHTML = a;

</script>

</body>

</html>

Copy

### ফলাফল

### = অপারেটর

32

"+=" এসাইনমেন্ট অপারেটরটি ভ্যারিয়েবলে নতুন মান যোগ করে।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>+= অপারেটর</h3>

<p id="test"></p>

<script>

var a = 32;

a += 8;

document.getElementById("test").innerHTML = a;

</script>

</body>

</html>

Copy

### ফলাফল

### += অপারেটর

40

"-" এসাইনমেন্ট অপারেটরটি ভ্যারিয়েবল থেকে ভ্যালু বিয়োগ করে।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>-= অপারেটর</h3>

<p id="test"></p>

<script>

var a = 32;

a -= 2;

document.getElementById("test").innerHTML = a;

</script>

</body>

</html>

Copy

### ফলাফল

### -= অপারেটর

30

"\*=" এসাইনমেন্ট অপারেটরটি ভ্যারিয়েবলকে গুণ করে।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>\*= অপারেটর</h3>

<p id="test"></p>

<script>

var a = 32;

a \*= 2;

document.getElementById("test").innerHTML = a;

</script>

</body>

</html>

Copy

### ফলাফল

### \*= অপারেটর

64

"/=" এসাইনমেন্ট অপারেটরটি ভ্যারিয়েবলকে ভাগ করে।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>/= অপারেটর</h3>

<p id="test"></p>

<script>

var a = 32;

a /= 2;

document.getElementById("test").innerHTML = a;

</script>

</body>

</html>

Copy

### ফলাফল

### /= অপারেটর

16

"%=" এসাইনমেন্ট অপারেটরটি ভাগশেষ জমা রাখে।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>%= অপারেটর</h3>

<p id="test"></p>

<script>

var a = 32;

a %= 10;

document.getElementById("test").innerHTML = a;

</script>

</body>

</html>

Copy

### ফলাফল

### %= অপারেটর

2Top of Form

# জাভাস্ক্রিপ্ট ফাংশন(Function)

জাভাস্ক্রিপ্ট ফাংশন হচ্ছে একগুচ্ছ কোড যা কোন নির্দিষ্ট কাজ সম্পাদন করার জন্য তৈরি করা হয়।

জাভাস্ক্রিপ্ট ফাংশন নিজে নিজে সম্পাদিত(execution) হয় না। এটা সম্পাদিত হয় যখন কোন কিছুর মাধ্যমে একে ডাকা হয়।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> এই উদাহারনে একটি ফাংশন কল করে যা একটি ক্যালকুলেশন এক্সিকিউট করে এবং নিম্নোক্ত ফলাফল প্রদান করেঃ </h4>

<p id="test"></p>

<script>

function myFunc(a, b) {

return a \* b;

}

document.getElementById("test").innerHTML = myFunc(5, 2);

</script>

</body>

</html>

Copy

### ফলাফল

#### এই উদাহারনে একটি ফাংশন কল করে যা একটি ক্যালকুলেশন এক্সিকিউট করে এবং নিম্নোক্ত ফলাফল প্রদান করেঃ

10

## জাভাস্ক্রিপ্ট ফাংশনের গঠনপ্রণালী(Syntax)

জাভাস্ক্রিপ্ট ফাংশন তৈরি করতে প্রথমে function কিওয়ার্ড(keyword) লিখতে হবে, তারপর ফাংশনের নাম এবং এর ডান পাশে () এবং এর ডানপাশে {} লিখতে হয়।

ফাংশনের নাম সাধারনত অক্ষর(letters),সংখ্যা(digits),আন্ডারস্কোর(\_), ডলার($) চিহ্ন দিয়ে লেখা যায়।

প্রথম বন্ধনীর মাঝের প্যারামিটারগুলো কমার(,) মাধ্যমে আলাদা করা হয়ঃ (**parameter1,  parameter2, ...**)

দ্বিতীয় বন্ধনীর {} মধ্যে অবস্থিত কোডগুলো ফাংশনকে কল করার মাধ্যমে সম্পাদিত হয়।

function name(parameter1, parameter2, parameter3) {

সম্পাদনযোগ্য কোড

}

Copy

ফাংশন প্যারামিটার(parameter) হচ্ছে ফাংশন ডেফিনেশনের প্রথম বন্ধনীর মধ্যে ব্যবহৃত নাম।

ফাংশন আর্গুমেন্ট(arguments) হচ্ছে ফাংশনকে ডাকার(invoke) সময় প্যারামিটারের মান যেগুলো ফাংশন ব্যবহার করে।

ফাংশনের মধ্যে আর্গুমেন্টগুলো লোকাল ভ্যারিয়েবলের(local variables) মত আচরণ করে।

|  |  |
| --- | --- |
|  | জাভাস্ক্রিপ্টের ফাংশনগুলো অন্যান্য প্রোগ্রামিং ল্যাঙ্গুয়েজের প্রসিডিউর(Procedure) অথবা সাব্রুটিনের(Subroutine) মত। |

## ফাংশন ডাকা(Invocation)

কোন কিছুর মাধ্যমে ফাংশনকে ডাক(Call) দেওয়া হলে ফাংশনের ভিতরের কোডগুলো সম্পাদিত(Execution) হয়ঃ

* যখন কোন ইভেন্ট ঘটে (ব্যবহারকারী বাটনে ক্লিক করলে)
* যখন একে ডাকা(Call) হয়
* স্বয়ংক্রিয়ভাবে (নিজেই নিজেকে কল করে)

এই টিউটোরিয়ালের পরবর্তী পরিচ্ছেদে ফাংশন কল সম্পর্কে আপনি আরো শিখবেন।

## ফাংশন রিটার্ন(Return)

জাভাস্ক্রিপ্ট রিটার্ন(return) স্টেটমেন্টের কাছে পৌঁছালে ফাংশন এক্সিকিউশন বন্ধ হয়ে যায়।

যদি ফাংশনকে কোন স্টেটমেন্টে কল করা হয় তাহলে জাভাস্ক্রিপ্ট ঐ স্টেটমেন্টের পরে সেই ফাংশনের কোড করার জন্য ফেরত যায়।

ফাংশন প্রায়ই রিটার্ন(return) ভ্যালু এক্সিকিউট করে। রিটার্নকৃত ভ্যালু কলারের(Caller) কাছে ফেরত পাঠানো হয়ঃ

### উদাহরণ

দুইটি সংখ্যার গুণফল রিটার্ন করিঃ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> এই উদাহারনে একটি ফাংশন কল করে যা একটি ক্যালকুলেশন এক্সিকিউট করে এবং নিম্নোক্ত ফলাফল প্রদান করেঃ </h4>

<p id="test"></p>

<script>

function myFunc(a, b) {

return a \* b;

}

document.getElementById("test").innerHTML = myFunc(5, 2);

</script>

</body>

</html>

Copy

### ফলাফল

#### এই উদাহারনে একটি ফাংশন কল করে যা একটি ক্যালকুলেশন এক্সিকিউট করে এবং নিম্নোক্ত ফলাফল প্রদান করেঃ

10

## ফাংশন কেন?

আপনি কোড বারবার ব্যবহার করতে পারবেনঃ কোড একবার লিখে অসংখ্যবার ব্যবহার করতে পারবেন।

ভিন্ন ভিন্ন ফলাফল পাওয়ার জন্য আপনি বিভিন্ন আর্গুমেন্ট ব্যবহার করে একই কোড অনেকবার ব্যবহার করতে পারেন।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> এই উদাহারনে একটি ফাংশন ভিন্ন ভিন্ন আর্গুমেন্ট দ্বারা কল করে যা একাধিক ক্যালকুলেশন এক্সিকিউট করে এবং নিম্নোক্ত ফলাফল প্রদান করেঃ </h4>

<p id="test"></p>

<script>

function myFunc(a, b) {

return a \* b;

}

document.getElementById("test").innerHTML = myFunc(5, 2) + "<br>" + myFunc(6, 4);

</script>

</body>

</html>

Copy

### ফলাফল

#### এই উদাহারনে একটি ফাংশন ভিন্ন ভিন্ন আর্গুমেন্ট দ্বারা কল করে যা একাধিক ক্যালকুলেশন এক্সিকিউট করে এবং নিম্নোক্ত ফলাফল প্রদান করেঃ

10  
24

## () অপারেটর ফাংশনকে কল করে

উপরোক্ত উদাহরনে ব্যবহৃত toCelsius দ্বারা ফাংশন অবজেক্টকে বুঝা‍য় এবং toCelsius() দ্বারা ফাংশনের ফলাফলকে বুঝায়।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> () ছাড়া ফাংশন এক্সেস করলে ফাংশন ডেফিনেশন রিটার্ন হয়ঃ </h4>

<p id="test"></p>

<script>

function myFunc(a, b) {

return a \* b;

}

document.getElementById("test").innerHTML = myFunc;

</script>

</body>

</html>

Copy

### ফলাফল

#### () ছাড়া ফাংশন এক্সেস করলে ফাংশন ডেফিনেশন রিটার্ন হয়ঃ

function myFunc(a, b) { return a \* b; }

## ফাংশনকে ভ্যারিয়েবল হিসেবে ব্যবহার

জাভাস্ক্রিপ্টে আপনি ফাংশনকে ভ্যারিয়েবলের মত ব্যবহার করতে পারেন।

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

function myFunc(a, b) {

return a \* b;

}

var result = myFunc(5, 2);

document.getElementById("test").innerHTML = "ফলাফল হলো " + result ;

</script>

</body>

</html>

Copy

একটি ফাংশনের রিটার্ন ভ্যালু একটি ভ্যারিয়েবলে জমা না রেখেঃ

আপনি ফাংশনকে সরাসরি ভ্যারিয়েবলের মত ব্যবহার করতে পারেনঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

function myFunc(a, b) {

return a \* b;

}

document.getElementById("test").innerHTML = "ফলাফল হলো " + myFunc(5, 2) ;

</script>

</body>

</html>

Copy

### ফলাফল

ফলাফল হলো 10

# জাভাস্ক্রিপ্ট অবজেক্ট(Object)

## বাস্তব জীবনের অবজেক্ট, প্রোপার্টি, এবং মেথড

বাস্তব জীবনে, একটি গাড়ি হল অবজেক্ট। একটি গাড়ির ওজন এবং রং প্রোপার্টি, মেথড হচ্ছে গাড়ি চালু করা বা বন্ধ করাঃ

|  |  |  |
| --- | --- | --- |
| অবজেক্ট | প্রোপার্টি | মেথড |
| https://www.sattacademy.com/images/objectExplained.gif | car.name = Lamborghini  car.model = 500  car.weight = 850kg  car.color = white | car.start()  car.drive()  car.brake()  car.stop() |

সব গাড়ির একই ধরনের প্রোপার্টি থাকে, কিন্তু বিভিন্ন গাড়ির বিভিন্ন ধরনের প্রোপার্টি ভ্যালু থাকে।

সব গাড়ির একই ধরনের মেথড থাকে, কিন্তু মেথডগুলো বিভিন্ন সময়ে সম্পাদন হয়।

## জাভাস্ক্রিপ্ট অবজেক্ট

আপনারা ইতিমধ্যে জেনেছেন যে, জাভাস্ক্রিপ্ট ভ্যারিয়েবলগুলো হলো ডাটার কন্টেইনার বা পাত্র।

এই কোডে car নামের একটি ভ্যারিয়েবলে একটি সাধারণ ভ্যালু এসাইন করা হয়েছেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>একটি জাভাস্ক্রিপ্ট ভেরিয়েবল তৈরি </h4>

<p id="test"></p>

<script>

var person = "Tamim";

document.getElementById("test").innerHTML = person;

</script>

</body>

</html>

Copy

### ফলাফল

#### একটি জাভাস্ক্রিপ্ট ভেরিয়েবল তৈরি

Tamim

অবজেক্টও এক ধরনের ভ্যারিয়েবল। কিন্তু অবজেক্ট এক সাথে অনেক ভ্যালু ধারন করতে পারে।

এই কোডে member নামের ভ্যারিয়েবলে একসাথে অনেক ভ্যালু(Azizur, Rahman, 32, black) এসাইন করা হয়েছেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>একটি জাভাস্ক্রিপ্ট অবজেক্ট তৈরি </h4>

<p id="test"></p>

<script>

var member = {firstName:"Azizur", lastName:"Rahman", age:32, eyeColor:"black"};

document.getElementById("test").innerHTML = member.eyeColor;

</script>

</body>

</html>

Copy

### ফলাফল

#### একটি জাভাস্ক্রিপ্ট অবজেক্ট তৈরি

black

ভ্যালুগুলো name:value আকারে জোড়ায় জোড়ায় লিখা হয়েছে (name and value কোলন দ্বারা আলাদা করা হয়েছে)।

জাভাস্ক্রিপ্ট অবজেক্ট নামযুক্ত ভ্যালু(eyeColor:"black") ধারন করে।

## অবজেক্ট প্রোপার্টি

name:value এর জোড়াকে জাভাস্ক্রিপ্ট অবজেক্টে প্রোপার্টি বলা হয়।

var member = {firstName:"Azizur", lastName:"Rahman", age:32, eyeColor:"black"};

Copy

|  |  |
| --- | --- |
| **প্রোপার্টি** | **প্রোপার্টি ভ্যালু** |
| firstName | Azizur |
| lastName | Rahman |
| age | 50 |
| eyeColor | black |

## অবজেক্ট মেথড

অবজেক্টের উপর সম্পাদিত কাজকেই মেথড বলে।

মেথডকে ফাংশন ডেফিনেশন আকারে প্রোপার্টির মধ্যে জমা রাখতে হয়।

|  |  |
| --- | --- |
| **প্রোপার্টি** | **প্রোপার্টি ভ্যালু** |
| firstName | Azizur |
| lastName | Rahman |
| age | 50 |
| eyeColor | black |
| fullName | function() {return this.firstName + " " + this.lastName;} |

জাভাস্ক্রিপ্ট অবজেক্ট হলো প্রোপার্টি এবং মেথডের কন্টেইনার।

## অবজেক্টের সংজ্ঞা

অবজেক্ট লিটারালের(object literal) মাধ্যমে আপনি একটি জাভাস্ক্রিপ্ট অবজেক্ট তৈরি করতে পারেনঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>একটি জাভাস্ক্রিপ্ট অবজেক্ট তৈরি </h4>

<p id="test"></p>

<script>

var member = {firstName:"Azizur", lastName:"Rahman", age:32, eyeColor:"black"};

document.getElementById("test").innerHTML =

member.firstName + " " + member.lastName + " " + "এর বয়স" + " " + member.age + " " + " বছর।";

</script>

</body>

</html>

Copy

### ফলাফল

#### একটি জাভাস্ক্রিপ্ট অবজেক্ট তৈরি

Azizur Rahman এর বয়স 32 বছর।

অবজেক্ট ডেফিনেশনে স্পেস এবং লাইন ব্রেক(line break) গুরুত্বপূর্ণ নয়। একটি অবজেক্ট ডেফিনেশনকে একাধিক লাইনে লেখা যেতে পারেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>একটি জাভাস্ক্রিপ্ট অবজেক্ট তৈরি </h4>

<p id="test"></p>

<script>

var member = {

firstName:"Azizur",

lastName:"Rahman",

age:32,

eyeColor:"black"

};

document.getElementById("test").innerHTML =

member.firstName + " " + member.lastName + " " + "এর বয়স" + " " + member.age + " " + " বছর।";

</script>

</body>

</html>

Copy

### ফলাফল

#### একটি জাভাস্ক্রিপ্ট অবজেক্ট তৈরি

Azizur Rahman এর বয়স 32 বছর।

## অবজেক্টের প্রোপার্টি এক্সেস

আপনি দুইভাবে অবজেক্টের প্রোপার্টিকে এক্সেস করতে পারেনঃ

objectName.propertyName

Copy

অথবা

objectName["propertyName"]

Copy

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> একটি অবজেক্ট প্রোপার্টি একসেস করার দুইটি উপায় আছেঃ

</h4>

<h4>আপনি member.property অথবা member["property"]

ব্যবহার করতে পারেন। </p>

<p id="test"></p>

<script>

var member = {firstName:"Azizur", lastName:"Rahman", age:32, eyeColor:"black"};

document.getElementById("test").innerHTML = member.eyeColor;

</script>

</body>

</html>

Copy

### ফলাফল

#### একটি অবজেক্ট প্রোপার্টি একসেস করার দুইটি উপায় আছেঃ

#### আপনি member.property অথবা member["property"] ব্যবহার করতে পারেন।

**black**

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> একটি অবজেক্ট প্রোপার্টি একসেস করার দুইটি উপায় আছেঃ

</h4>

<h4>আপনি member.property অথবা member["property"]

ব্যবহার করতে পারেন। </p>

<p id="test"></p>

<script>

var member = {firstName:"Azizur", lastName:"Rahman", age:32, eyeColor:"black"};

document.getElementById("test").innerHTML = member["eyeColor"];

</script>

</body>

</html>

Copy

### ফলাফল

#### একটি অবজেক্ট প্রোপার্টি একসেস করার দুইটি উপায় আছেঃ

#### আপনি member.property অথবা member["property"] ব্যবহার করতে পারেন।

**black**

## অবজেক্টের মেথড এক্সেস

আপনি নিম্নলিখিত পদ্ধতিতে অবজেক্টের মেথডকে এক্সেস করতে পারবেনঃ

objectName.methodName()

Copy

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>একটি অবজেক্ট মেথড তৈরি এবং ব্যবহার।

</h4>

<h4>ফাংশন ডেফিনেশন হচ্ছে একটি অবজেক্ট মেথড যা প্রোপার্টি ভ্যালু হিসেবে ধারন করে। </p>

<p id="test"></p>

<script>

var member = {firstName:"Azizur", lastName:"Rahman", age:32, eyeColor:"black", fullName : function() {

return this.firstName + " " + this.lastName;

}};

document.getElementById("test").innerHTML = member.fullName();

</script>

</body>

</html>

Copy

### ফলাফল

#### একটি অবজেক্ট মেথড তৈরি এবং ব্যবহার।

#### ফাংশন ডেফিনেশন হচ্ছে একটি অবজেক্ট মেথড যা প্রোপার্টি ভ্যালু হিসেবে ধারন করে।

**Azizur Rahman**

যদি আপনি fullName মেথডকে () ছাড়া এক্সেস করতে চান তবে এটি আপনাকে ফাংশন ডেফিনেশনকে হুবুহু রিটার্ন করবেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>একটি অবজেক্ট মেথড হচ্ছে একটি ফাংশন ডেফিনেশন যা প্রোপার্টি ভ্যালু হিসেবে জমা থাকে।

</h4>

<h4>আপনি যদি () ছাড়া একসেস করেন, তবে ফাংশন ডেফিনেশন রিটার্ন করতে পারবেনঃ </p>

<p id="test"></p>

<script>

var member = {firstName:"Azizur", lastName:"Rahman", age:32, eyeColor:"black", fullName : function() {

return this.firstName + " " + this.lastName;

}};

document.getElementById("test").innerHTML = member.fullName;

</script>

</body>

</html>

Copy

### ফলাফল

#### একটি অবজেক্ট মেথড হচ্ছে একটি ফাংশন ডেফিনেশন যা প্রোপার্টি ভ্যালু হিসেবে জমা থাকে।

#### আপনি যদি () ছাড়া একসেস করেন, তবে ফাংশন ডেফিনেশন রিটার্ন করতে পারবেনঃ

**function() { return this.firstName + " " + this.lastName; }**

একটি প্রোপার্টির ভ্যালু হিসেবে যখন ফাংশন ডেফিনেশন রাখা হয় তখন ঐ প্রোপার্টিকে মেথড বলা হয়।

## কখনোই Strings, Numbers, এবং Booleans কে অবজেক্ট হিসেবে ডিক্লেয়ার করা উচিত না!

জাভাস্ক্রিপ্টে একটি ভ্যারিয়েবলকে "new" কিওয়ার্ড দ্বারা ডিক্লেয়ার করলে ঐ ভ্যারিয়েবলটি একটি অবজেক্ট হয়ে যায়ঃ

var a = new String(); // এখানে a কে String অবজেক্ট হিসেবে ডিক্লেয়ার করা হয়েছে

var b = new Number(); // এখানে b কে Number অবজেক্ট হিসেবে ডিক্লেয়ার করা হয়েছে

var c = new Boolean(); // এখানে c কে Boolean অবজেক্ট হিসেবে ডিক্লেয়ার করা হয়েছে

Copy

String, Number এবং Boolean অবজেক্ট তৈরি করা থেকে বিরত থাকুন। এইগুলো আপনার কোডকে জটিল এবং সম্পাদনের গতি কমিয়ে দেয়।

# জাভাস্ক্রিপ্ট স্কোপ

স্কোপ হলো আপনি এক্সেস করতে পারেন এমন ভ্যারিয়েবলের সেট।

## জাভাস্ক্রিপ্ট স্কোপ

জাভাস্ক্রিপ্টে অবজেক্ট এবং ফাংশনও ভ্যারিয়েবল।

আপনি এক্সেস করতে পারেন এমন ভ্যারিয়েবল, অবজেক্ট এবং ফাংশনের সেটকে জাভাস্ক্রিপ্টে স্কোপ বলা হয়।

জাভাস্ক্রিপ্টের ফাংশন স্কোপ রয়েছেঃ ফাংশনের মধ্যে স্কোপ পরিবর্তিত হয়।

## জাভাস্ক্রিপ্ট লোকাল ভ্যারিয়েবল

জাভাস্ক্রিপ্ট ফাংশনের মধ্যে যে ভ্যারিয়েবল বা চলক ডিক্লেয়ার করা হয় তাকে লোকাল ভ্যারিয়েবল বলে।

লোকাল ভ্যারিয়েবলের লোকাল স্কোপ থাকেঃ যা শুধুমাত্র ফাংশনের মধ্যেই এক্সেস করা সম্ভব।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>লোকাল ভেরিয়েবল animal ফাংশনের বাইরে এক্সেস করা যাবে না। </h4>

<p id="test"></p>

<script>

function myFunc() {

var animal = "Tiger";

}

myFunc();

document.getElementById("test").innerHTML = "প্রানীটি হছে " + animal;

</script>

</body>

</html>

Copy

### ফলাফল

#### লোকাল ভেরিয়েবল animal ফাংশনের বাইরে এক্সেস করা যাবে না।

যেহেতু লোকাল ভ্যারিয়েবল শুধুমাত্র একটি ফাংশনের ভিতরে এক্সেস করা যায়, তাই একই নামের ভ্যারিয়েবল বিভিন্ন ফাংশনের ভিতরে ব্যবহার করা যেতে পারে।

যখন ফাংশনকে কল করা হয় তখন লোকাল ভ্যারিয়েবল তৈরি হয় এবং ফাংশনের কাজ সম্পন্ন হলে লোকাল ভ্যারিয়েবলগুলো ডিলেট হয়ে যায়।

## জাভাস্ক্রিপ্ট গ্লোবাল ভ্যারিয়েবল

জাভাস্ক্রিপ্ট ফাংশনের বাইরে যে ভ্যারিয়েবলকে ডিক্লেয়ার করা হয় তাকে গ্লোবাল ভ্যারিয়েবল বলে।

গ্লোবাল ভ্যারিয়েবলের গ্লোবাল স্কোপ থাকেঃ ওয়েব পেজের সকল স্ক্রিপ্ট এবং ফাংশন একে এক্সেস করতে পারে।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> গ্লোবাল ভেরিয়েবল যে কোন স্ক্রিপ্ট বা ফাংশনেে এক্সেস করা যাবে।</h4>

<p id="test"></p>

<script>

var animal = "Tiger";

function myFunc() {

document.getElementById("test").innerHTML = "প্রানীটি হছে " + animal;

}

myFunc();

</script>

</body>

</html>

Copy

### ফলাফল

#### গ্লোবাল ভেরিয়েবল যে কোন স্ক্রিপ্ট বা ফাংশনেে এক্সেস করা যাবে।

প্রানীটি হছে Tiger

## স্বয়ংক্রিয়ভাবে গ্লোবাল

যদি আপনি কোন ভ্যারিয়েবলকে ডিক্লেয়ার না করে ভ্যালু এসাইন করেন তাহলে ভ্যারিয়েবলটি সয়ংক্রিয়ভাবে একটি **গ্লোবাল** ভ্যারিয়েবলে পরিণত হবে।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>ডিক্লেয়ার করা ছাড়াই যদি আপনি ভেরিয়েবলে একটি ভ্যালু এসাইন করেন,এটা স্বয়ংক্রিয়ভাবে GLOBAL-এ হয়ে যাবেঃ </h4>

<p id="test"></p>

<script>

function myFunc() {

animal = "Tiger";

}

myFunc();

document.getElementById("test").innerHTML = "প্রানীটি হছে " + animal;

</script>

</body>

</html>

Copy

### ফলাফল

#### ডিক্লেয়ার করা ছাড়াই যদি আপনি ভেরিয়েবলে একটি ভ্যালু এসাইন করেন,এটা স্বয়ংক্রিয়ভাবে GLOBAL-এ হয়ে যাবেঃ

প্রানীটি হছে Tiger

প্রয়োজন ছাড়া গ্লোবাল ভ্যারিয়েবল তৈরি না করাই উত্তম।

"Strict Mode" এ স্বয়ংক্রিয়ভাবে গ্লোবাল ভ্যারিয়েবলকে এড়িয়ে চলা হয়।

## এইচটিএমএলে গ্লোবাল ভ্যারিয়েবল

জাভাস্ক্রিপ্টে গ্লোবাল স্কোপ সম্পূর্ণ জাভাস্ক্রিপ্টকে বুঝায়।

এইচটিএমএলে উইন্ডো অবজেক্ট হচ্ছে গ্লোবাল স্কোপ। সকল গ্লোবাল ভ্যারিয়েবল উইন্ডো অবজেক্টে অন্তর্গত থাকে।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>এইচটিএমএলে, সকল global ভেরিয়েবল window ভেরিয়েবলে পরিনত হবে।</h4>

<p id="test"></p>

<script>

var animal = "Tiger";

document.getElementById("test").innerHTML = "প্রানীটি হছে " + window.animal;

</script>

</body>

</html>

Copy

### ফলাফল

#### এইচটিএমএলে, সকল global ভেরিয়েবল window ভেরিয়েবলে পরিনত হবে।

প্রানীটি হছে Tiger

## আপনি জানেন কী?

গ্লোবাল ভ্যারিয়েবল(বা ফাংশন) উইন্ডো ভ্যারিয়েবলকে(বা ফাংশন) মুছে ফেলতে পারে।  
উইন্ডো অবজেক্টসহ যেকোন ফাংশন, আপনার গ্লোবাল ভ্যারিয়েবল এবং ফাংশনকে মুছে ফেলতে পারে।

## জাভাস্ক্রিপ্ট ভ্যারিয়েবলের জীবনকাল

জাভাস্ক্রিপ্ট ভ্যারিয়েবল ডিক্লেয়ার করা হলে এর জীবনকাল শুরু হয়।

ফাংশনের কার্য সম্পন্ন হলে লোকাল ভ্যারিয়েবল মুছে যায়।

ওয়েব পেজ বন্ধ করলে গ্লোবাল ভ্যারিয়েবল মুছে যায়।

## ফাংশন আর্গুমেন্ট

ফাংশন আর্গুমেন্ট(প্যারামিটার) ফাংশনের ভিতরে লোকাল ভ্যারিয়েবল হিসেবে কাজ করে।

# জাভাস্ক্রিপ্ট ইভেন্ট

এইচটিএমএল এলিমেন্টে ঘটে এমন সকল কিছুকেই ইভেন্ট বলা হয়।

এইচটিএমএল পেজে জাভাস্ক্রিপ্ট ব্যবহার করা হলে জাভাস্ক্রিপ্ট ঐ সকল ইভেন্টে "প্রতিক্রিয়া" করতে পারে।

## এইচটিএমএল ইভেন্ট

ব্রাউজার অথবা ইউজার যা কিছু করে তা এইচটিএমএল ইভেন্ট হতে পারে।

এখানে এইচটিএমএল ইভেন্টের কিছু উদাহরণ দেওয়া হলঃ

* একটি এইচটিএমএল ওয়েব পেজ লোডিং সম্পন্ন হওয়া
* একটি এইচটিএমএল ইনপুট ফিল্ড পরিবর্তন করা
* একটি এইচটিএমএল বাটনে ক্লিক করা

কোন ইভেন্ট ঘটলে আপনি চাইবেন কিছু ঘটুক।

কোন ইভেন্ট ঘটলে জাভাস্ক্রিপ্ট আপনাকে কোড সম্পাদন করতে সহায়তা করবে।

এইচটিএমএল আমাদেরকে জাভাস্ক্রিপ্ট কোডের মাধ্যমে এইচটিএমএল এলিমেন্টে ইভেন্ট হ্যান্ডেলার এট্রিবিউট যোগ করতে সাহায্য করে।

সিঙ্গেল কোটেশন দিয়েঃ

<HTML-element event='JavaScript'>

Copy

ডাবল কোটেশন দিয়েঃ

<HTML-element event="JavaScript">

Copy

নিচের উদাহরণে, একটি বাটন এলিমেন্টে onclick এট্রিবিউট যোগ করে দেখানো হলোঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<button onclick="document.getElementById('test').innerHTML= new Date().getFullYear()">Year?</button>

<p id="test"></p>

</body>

</html>

### ফলাফল

### Year?

2018

উপরের উদাহরণে, জাভাস্ক্রিপ্ট কোড id="test" যুক্ত এলিমেন্টের কন্টেন্টকে পরিবর্তন করেছে।

পরবর্তী উদাহরণে, কোড(this.innerHTML ব্যবহার করে) নিজ এলিমেন্টের কন্টেন্টকে পরিবর্তন করবেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<button onclick="this.innerHTML= new Date().getFullYear()">Year?</button>

</body>

</html>

### ফলাফল

2018

জাভাস্ক্রিপ্ট কোড বেশিরভাগ ক্ষেত্রে কয়েক লাইন দীর্ঘ হতে পারে। তাই ইভেন্ট এট্রিবিউটে ফাংশনকে কল করা যায়ঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<button onclick="currentYear()">Year?</button>

<p id="test"></p>

<script>

function currentYear(){

document.getElementById('test').innerHTML= new Date().getFullYear();

}

</script>

</body>

</html>

Copy

### ফলাফল

2018

## এইচটিএমএল ইভেন্ট

এখানে কিছু এইচটিএমএল ইভেন্টের তালিকা দেয়া হলোঃ

|  |  |
| --- | --- |
| **ইভেন্ট** | **বিবরণ** |
| onchange | একটি এইচটিএমএল এলিমেন্ট পরিবর্তন করা হয়েছে |
| onclick | ব্যবহারকারী একটি এইচটিএমএল এলিমেন্ট ক্লিক করেছে |
| onmouseover | ব্যবহারকারী একটি এইচটিএমএল এলিমেন্টের উপরে মাউস নিয়েছে |
| onmouseout | ব্যবহারকারী একটি এইচটিএমএল এলিমেন্টের উপর থেকে মাউস সরিয়ে নিলে |
| onkeydown | ব্যবহারকারী একটি কীবোর্ড কী চাপলে |
| onload | ব্রাউজার পেজ লোড সম্পন্ন করলে |

তালিকাটি অনেক বড়। আমাদের [জাভাস্ক্রিপ্ট রেফারেন্স এইচটিএমএল ডোম ইভেন্ট](https://www.sattacademy.com/jsref/dom_obj_event.php) দেখুন।

## এইচটিএমএল ডোম ইভেন্ট

এইচটিএমএল ডোম ইভেন্ট একটি এইচটিএমএল ডকুমেন্ট এলিমেন্টের বিভিন্ন ইভেন্ট হ্যান্ডেলার জাভাস্ক্রিপ্টে রেজিস্টার করতে অনুমতি দেয়।

ইভেন্ট সাধারনত ফাংশনের সাথে সমন্বয় করার জন্য ব্যবহার হয় এবং ফাংশনটির ইভেন্ট ঘটার আগে এক্সিকিউট হবে না।( উদাহরণস্বরুপ একজন ইউজার যখন একটি বাটনে ক্লিক করে।)

## এইচটিএমএল ডোম ইভেন্ট

## মাউস ইভেন্ট

|  |  |  |
| --- | --- | --- |
| **ইভেন্ট** | **বর্ণনা** | **ডোম** |
| [onclick](https://www.sattacademy.com/jsref/event_onclick.php) | ইউজার একটি এলিমেন্টে যখন ক্লিক করে তখন এই ঘটনা ঘটে। | ২ |
| [oncontextmenu](https://www.sattacademy.com/jsref/event_oncontextmenu.php) | একজন ইউজার যখন একটি কন্টেক্স মেনু ওপেন করার জন্য right-click করে তখন এই ঘটনা ঘটে। | ৩ |
| [ondblclick](https://www.sattacademy.com/jsref/event_ondblclick.php) | ইউজার যখন একটি এলিমেন্টের উপর ডাবল ক্লিক করে তখন এই ঘটনা ঘটে। | ২ |
| [onmousedown](https://www.sattacademy.com/jsref/event_onmousedown.php) | ইউজার যখন মাউসকে একটি এলিমেন্টের উপর নিয়ে যায় তখন এই ঘটনা ঘটে। | ২ |
| [onmouseenter](https://www.sattacademy.com/jsref/event_onmouseenter.php) | পয়েন্টার যখন একটি এলিমেন্টের উপর move করা হয় তখন এই ঘটনা ঘটে। | ২ |
| [onmouseleave](https://www.sattacademy.com/jsref/event_onmouseleave.php) | পয়েন্টার যখন একটি এলিমেন্টের বাইরে নেয়া হয় তখন এই ঘটনা ঘটে। | ২ |
| [onmousemove](https://www.sattacademy.com/jsref/event_onmousemove.php) | পয়েন্টার যতক্ষন একটি এলিমেন্টের উপর ধরে রাখবে ততক্ষন এই ঘটনা ঘটে। | ২ |
| [onmouseover](https://www.sattacademy.com/jsref/event_onmouseover.php) | পয়েন্টার যখন একটি এলিমেন্টের উপর move করা হয় তখন এই ঘটনা ঘটে। | ২ |
| [onmouseout](https://www.sattacademy.com/jsref/event_onmouseout.php) | পয়েন্টার যখন একটি এলিমেন্টের বাইরে নেয়া হয় তখন এই ঘটনা ঘটে। | ২ |
| [onmouseup](https://www.sattacademy.com/jsref/event_onmouseup.php) | ইউজার যখন একটি এলিমেন্টের উপর থেকে মাউস বাটন সরিয়ে নেয় তখন এই ঘটনা ঘটে। | ২ |

## কিবোর্ড ইভেন্ট

|  |  |  |
| --- | --- | --- |
| **ইভেন্ট** | **Description** | **ডোম** |
| [onkeydown](https://www.sattacademy.com/jsref/event_onkeydown.php) | যখন ইউজার একটি key চাপ দেয় তখন এই ঘটনা ঘটে। | ২ |
| [onkeypress](https://www.sattacademy.com/jsref/event_onkeypress.php) | যখন ইউজার একটি key চাপ দেয় তখন এই ঘটনা ঘটে। | ২ |
| [onkeyup](https://www.sattacademy.com/jsref/event_onkeyup.php) | যখন ইউজার একটি key থেকে মাউস সরিয়ে নেয় তখন এই ঘটনা ঘটে। | ২ |

## ফ্রেম/অবজেক্ট ইভেন্ট

|  |  |  |
| --- | --- | --- |
| **ইভেন্ট** | **বর্ণনা** | **ডোম** |
| [onabort](https://www.sattacademy.com/jsref/event_onabort.php) | রিসোর্সের লোডিং যখন বাদ দেয়া হয় তখন এই ঘটনা ঘটে। | ২ |
| [onbeforeunload](https://www.sattacademy.com/jsref/event_onbeforeunload.php) | ডকুমেন্ট লোড হওয়ার আগে এই ঘটনাটি ঘটে। | ২ |
| [onerror](https://www.sattacademy.com/jsref/event_onerror.php) | যখন একটি অতিরিক্ত ফাইল লোডিং হয় তখন কোনো ভুল হলে এই ঘটনা ঘটে। | ২ |
| [onhashchange](https://www.sattacademy.com/jsref/event_onhashchange.php) | যখন একটি URL এর anchor এর একটি অংশ পরিবর্তন হয় তখন এই ঘটনা ঘটে। | ৩ |
| [onload](https://www.sattacademy.com/jsref/event_onload.php) | যখন একটি অবজেক্ট লোড হয় তখন এই ঘটনা ঘটে। | ২ |
| [onpageshow](https://www.sattacademy.com/jsref/event_onpageshow.php) | যখন ইউজার ওয়েবপেজে নেভিগেট করে তখন এই ঘটনা ঘটে। | ৩ |
| [onpagehide](https://www.sattacademy.com/jsref/event_onpagehide.php) | যখন ইউজার ওয়েবপেজের বাইরে নেভিগেট করে তখন এই ঘটনা ঘটে। | ৩ |
| [onresize](https://www.sattacademy.com/jsref/event_onresize.php) | যখন একটি ডকুমেন্টের ভিউ রিসাইজ করা হয় তখন এই ঘটনা ঘটে। | ২ |
| [onscroll](https://www.sattacademy.com/jsref/event_onscroll.php) | যখন এলিমেন্টের স্ক্রলবার স্ক্রল করা হয় তখন এই ঘটনা ঘটে। | ২ |
| [onunload](https://www.sattacademy.com/jsref/event_onunload.php) | একটি পেজ আনলোড থাকলে এই ঘটনা ঘটে।(<body> এর জন্য) | ২ |

## ফর্ম ইভেন্ট

|  |  |  |
| --- | --- | --- |
| **ইভেন্ট** | **বর্ণনা** | **ডোম** |
| [onblur](https://www.sattacademy.com/jsref/event_onblur.php) | একটি এলিমেন্ট যখন ফোকাস হারায় তখন এই ঘটনা ঘটে। | ২ |
| [onchange](https://www.sattacademy.com/jsref/event_onchange.php) | একটি ফরম এলিমেন্ট, সেলেকশন অথবা চেকের এলিমেন্ট যখন পরিবর্তন হয় তখন এই ঘটন ঘটে।( <input>, <keygen>, <select> এবং <textarea> এর জন্য ) | ২ |
| [onfocus](https://www.sattacademy.com/jsref/event_onfocus.php) | একটি এলিমেন্ট যখন ফোকাস পায় তখন এই ঘটনা ঘটে। | ২ |
| [onfocusin](https://www.sattacademy.com/jsref/event_onfocusin.php) | যখন একটি এলিমেণ্ট ফোকাস পায় তখন এই ঘটনা ঘটে। | ২ |
| [onfocusout](https://www.sattacademy.com/jsref/event_onfocusout.php) | যখন একটি এলিমেণ্ট ফোকাস হারায় তখন এই ঘটনা ঘটে। | ২ |
| [oninput](https://www.sattacademy.com/jsref/event_oninput.php) | যখন একটি এলিমেন্ট ইউজার ইনপুট পায় তখন এই ঘটনা ঘটে। | ৩ |
| [oninvalid](https://www.sattacademy.com/jsref/event_oninvalid.php) | যখন একটি এলিমেন্ট invalid হয় তখন এই ঘটনা ঘটে। | ৩ |
| [onreset](https://www.sattacademy.com/jsref/event_onreset.php) | যখন একটি reset দেওয়া হয় তখন এই ঘটনা ঘটে। | ২ |
| [onsearch](https://www.sattacademy.com/jsref/event_onsearch.php) | যখন ইউজার সার্চ ফিল্ডে কিছু লিখবে তখন এই ঘটনা ঘটে। ( <input="search"> এর জন্য) | ৩ |
| [onselect](https://www.sattacademy.com/jsref/event_onselect.php) | যখন ইউজার কিছু টেক্সট সেলেক্ট করবে তখন এই ঘটনা ঘটে। (<input> এবং <textarea> এর জন্য) | ২ |
| [onsubmit](https://www.sattacademy.com/jsref/event_onsubmit.php) | একটি ফরম যখন সাবমিট হয় তখন এই ঘটনা ঘটে। | ২ |

## ড্রাগ ইভেন্ট

|  |  |  |
| --- | --- | --- |
| **ইভেন্ট** | **বর্ণনা** | **ডোম** |
| [ondrag](https://www.sattacademy.com/jsref/event_ondrag.php) | যখন এলিমেন্ট ড্রাগ করা হয় তখন এই ঘটনা ঘটে। | ৩ |
| [ondragend](https://www.sattacademy.com/jsref/event_ondragend.php) | যখন ইউজার এলিমেন্টের ড্রাগ শেষ করে তখন এই ঘটনা ঘটে। | ৩ |
| [ondragenter](https://www.sattacademy.com/jsref/event_ondragenter.php) | যখন ড্রাগ করা এলিমেন্ট ড্রপ টার্গেটে যায় তখন এই ঘটনা ঘটে। | ৩ |
| [ondragleave](https://www.sattacademy.com/jsref/event_ondragleave.php) | যখন ড্রাগ করা এলিমেন্ট ড্রপ টার্গেট থেকে সরে যায় তখন এই ঘটনা ঘটে। | ৩ |
| [ondragover](https://www.sattacademy.com/jsref/event_ondragover.php) | যখন ড্রাগ করা এলিমেন্ট ড্রপ টার্গেটের উপরে থাকে তখন এই ঘটনা ঘটে। | ৩ |
| [ondragstart](https://www.sattacademy.com/jsref/event_ondragstart.php) | যখন ইউজার একটি এলিমেন্টে ড্রাগ করা শুরু করে তখন এই ঘটনা ঘটে। | ৩ |
| [ondrop](https://www.sattacademy.com/jsref/event_ondrop.php) | যখন ড্রাগ করা এলিমেন্ট ড্রপ টার্গেট থেকে ড্রপ হয় তখন এই ঘটনা ঘটে। | ৩ |

## ক্লিপবোর্ড ইভেন্ট

|  |  |  |
| --- | --- | --- |
| **ইভেন্ট** | **বর্ণনা** | **ডোম** |
| [oncopy](https://www.sattacademy.com/jsref/event_oncopy.php) | ইউজার যখন একটি এলিমেন্টের কন্টেন্ট কপি করে তখন এই ঘটনা ঘটে। |  |
| [oncut](https://www.sattacademy.com/jsref/event_oncut.php) | ইউজার যখন একটি এলিমেন্টের কন্টেন্ট কাট করে তখন এই ঘটনা ঘটে। |  |
| [onpaste](https://www.sattacademy.com/jsref/event_onpaste.php) | ইউজার যখন একটি এলিমেন্টের কন্টেন্ট পেস্ট করে তখন এই ঘটনা ঘটে। |  |

## প্রিন্ট ইভেন্ট

|  |  |  |
| --- | --- | --- |
| **ইভেন্ট** | **বর্ণনা** | **ডোম** |
| [onafterprint](https://www.sattacademy.com/jsref/event_onafterprint.php) | যখন একটি পেজ প্রিন্ট শুরু হয় অথবা যদি প্রিন্ট ডায়ালগ বক্স থেকে ক্লোজ করা হয় তখন এই ঘটনা ঘটে। | ৩ |
| [onbeforeprint](https://www.sattacademy.com/jsref/event_onbeforeprint.php) | যখন একটি পেজ প্রিন্টের কাছাকাছি যায় তখন এই ঘটনা ঘটে। | ৩ |

## মিডিয়া ইভেন্ট

|  |  |  |
| --- | --- | --- |
| **ইভেন্ট** | **বর্ণনা** | **ডোম** |
| [onabort](https://www.sattacademy.com/jsref/event_onabort_media.php) | যখন মিডিয়া লোডিং বাতিল হবে তখন এই ঘটনা ঘটে। | ৩ |
| [oncanplay](https://www.sattacademy.com/jsref/event_oncanplay.php) | যখন ব্রাউজার মিডিয়া চালু করতে পারে তখন এই ঘটনা ঘটে। (যখন এটি প্রথম থেকে যথেষ্ট বাফারিং হয়) | ৩ |
| [oncanplaythrough](https://www.sattacademy.com/jsref/event_oncanplaythrough.php) | যখন ব্রাউজার কোনো রকম কোনো বাফারিং ছাড়া চলতে পারে তখন এই ঘটনা ঘটে। | ৩ |
| [ondurationchange](https://www.sattacademy.com/jsref/event_ondurationchange.php) | যখন মিডিয়ার স্থিতিকাল পরিবর্তন হয় তখন এই ঘটনা ঘটে। | ৩ |
| onemptied | যখন কোনো খারাপ কিছু ঘটে এবং হঠাৎ মিডিয়া ফাইল unavailable হলে এই ঘটনা ঘটে। (যেমন অপ্রত্যাশিত কানেকশন চলে যাওয়া) | ৩ |
| [onended](https://www.sattacademy.com/jsref/event_onended.php) | যখন মিডিয়া শেষের দিকে চলে যায় তখন এই ঘটনা ঘটে। (মেসেজের জন্য বেশি গুরুত্বপূর্ণ যেমন "ধন্যবাদ শোনার জন্য ") | ৩ |
| [onerror](https://www.sattacademy.com/jsref/event_onerror_media.php) | যখন মিডিয়া ফাইল লোডিং হয় তখন যদি error হয় তখন এই ঘটনা ঘটে। | ৩ |
| [onloadeddata](https://www.sattacademy.com/jsref/event_onloadeddata.php) | মিডিয়া ডাটা লোড হলে এই ঘটনা ঘটে। | ৩ |
| [onloadedmetadata](https://www.sattacademy.com/jsref/event_onloadedmetadata.php) | যখন মেটা ডাটা লোড হয় তখন এই ঘটনা ঘটে।(যেমন ডাইমেনসন এবং ডিউরেশন) | ৩ |
| [onloadstart](https://www.sattacademy.com/jsref/event_onloadstart.php) | যখন ব্রাউজার নির্দিষ্ট মিডিয়া খুজে তখন এই ঘটনা ঘটে। | ৩ |
| [onpause](https://www.sattacademy.com/jsref/event_onpause.php) | যখন মিডিয়া ইউজার কর্তৃক বিরতি হয় তখন এই ঘটনা ঘটে। | ৩ |
| [onplay](https://www.sattacademy.com/jsref/event_onplay.php) | যখন মিডিয়া শুরু হয় অথবা বিরতি শেষ হয় তখন এই ঘটনা ঘটে। | ৩ |
| [onplaying](https://www.sattacademy.com/jsref/event_onplaying.php) | যখন মিডিয়া বিরতির পর চালু হয় অথবা বাফারিং এর পর শুরু হয় তখন এই ঘটনা ঘটে। | ৩ |
| [onprogress](https://www.sattacademy.com/jsref/event_onprogress.php) | ব্রাউজার যখন মিডিয়া ডাটা নিতে থাকে তখন এই ঘটনা ঘটে। | ৩ |
| [onratechange](https://www.sattacademy.com/jsref/event_onratechange.php) | যখন মিডিয়ার স্পিড পরিবর্তন হয় তখন এই ঘটনা ঘটে। | ৩ |
| [onseeked](https://www.sattacademy.com/jsref/event_onseeked.php) | ইউজার যখন মিডিয়া moving/skipping বন্ধ করে তখন এই ঘটনা ঘটে। | ৩ |
| [onseeking](https://www.sattacademy.com/jsref/event_onseeking.php) | ইউজার যখন মিডিয়া moving/skipping শুরু করে তখন এই ঘটনা ঘটে। | ৩ |
| [onstalled](https://www.sattacademy.com/jsref/event_onstalled.php) | যখন ব্রাউজার মিডিয়া ডাটা পাওয়ার চেষ্টা করে কিন্তু ডাটা available না থাকলে এই ঘটনা ঘটে। | ৩ |
| [onsuspend](https://www.sattacademy.com/jsref/event_onsuspend.php) | যখন ব্রাউজার ইচ্ছাকৃতভাবে মিডিয়া ডাটা না নিলে তখন এই ঘটনা ঘটে। | ৩ |
| [ontimeupdate](https://www.sattacademy.com/jsref/event_ontimeupdate.php) | যখন প্লে-এর পজিশন পরিবর্তন হয় তখন এই ঘটনা ঘটে। | ৩ |
| [onvolumechange](https://www.sattacademy.com/jsref/event_onvolumechange.php) | যখন মিডিয়ার ভলিউম পরিবর্তন করা হয় তখন এই ঘটনা ঘটে। | ৩ |
| [onwaiting](https://www.sattacademy.com/jsref/event_onwaiting.php) | যখন মিডিয়া বিরতি দেয়া হয় তখন এই ঘটনা ঘটে।(যেমন যখন মিডিয়া অধিক ডাটার ক্ষেত্রে বাফার হয়)। | ৩ |

## অ্যানিমেশন ইভেন্ট

|  |  |  |
| --- | --- | --- |
| **ইভেন্ট** | **বর্ণনা** | **ডোম** |
| [animationend](https://www.sattacademy.com/jsref/event_animationend.php) | যখন সিএসএস অ্যানিমেশন সম্পূর্ণ হয় তখন এই ঘটনা ঘটে। | ৩ |
| [animationiteration](https://www.sattacademy.com/jsref/event_animationiteration.php) | যখন সিএসএস অ্যানিমেশন পূনরাবৃত্তি হয় তখন এই ঘটনা ঘটে। | ৩ |
| [animationstart](https://www.sattacademy.com/jsref/event_animationstart.php) | যখন সিএসএস অ্যানিমেশন শুরু হয় তখন এই ঘটনা ঘটে। | ৩ |

## ট্রাঞ্জিশন ইভেন্ট

|  |  |  |
| --- | --- | --- |
| **ইভেন্ট** | **বর্ণনা** | **ডোম** |
| [transitionend](https://www.sattacademy.com/jsref/event_transitionend.php) | যখন সিএসএস ট্রাঞ্জিশন সম্পূর্ণ হয় তখন এই ঘটনা ঘটে। | ৩ |

## সার্ভার-সেন্ট ইভেন্ট

|  |  |  |
| --- | --- | --- |
| **ইভেন্ট** | **বর্ণনা** | **ডোম** |
| [onerror](https://www.sattacademy.com/jsref/event_onerror_sse.php) | ইভেন্ট সোর্স এর সাথে যখন কোনো ভুল হয় তখন এই ঘটনা ঘটে। |  |
| [onmessage](https://www.sattacademy.com/jsref/event_onmessage_sse.php) | ইভেন্ট সোর্স এর সময় যখন একটি মেসেজ পাওয়া যায় তখন এই ঘটনা ঘটে। |  |
| [onopen](https://www.sattacademy.com/jsref/event_onopen_sse.php) | ইভেন্ট সোর্স ওপেন এর সময় যখন একটি কানেকশন থাকে তখন এই ঘটনা ঘটে। |  |

## মিস্ক ইভেন্ট

|  |  |  |
| --- | --- | --- |
| **ইভেন্ট** | **বর্ণনা** | **ডোম** |
| onmessage | একটি অবজেক্ট থেকে যখন একটি মেসেজ পাওয়া যায় তখন এই ঘটনা ঘটে। | ৩ |
| onmousewheel | বাতিল। এর পরিবর্তে [onwheelইভেন্ট ব্যবহার করুন।](https://www.sattacademy.com/jsref/event_onwheel.php) |  |
| [ononline](https://www.sattacademy.com/jsref/event_ononline.php) | ব্রাউজার যখন অনলাইনে কাজ শুরু করে তখন এই ঘটনা ঘটে। | ৩ |
| [onoffline](https://www.sattacademy.com/jsref/event_onoffline.php) | ব্রাউজার যখন অফলাইনে কাজ শুরু করে তখন এই ঘটনা ঘটে। | ৩ |
| onpopstate | যখন উইন্ডোর হিস্টোরি পরিবর্তন হয় তখন এই ঘটনা ঘটে। | ৩ |
| [onshow](https://www.sattacademy.com/jsref/event_onshow.php) | যখন একটি <menu>এলিমেন্ট কন্টেক্সট মেনুর মত দেখায় তখন এই ঘটনা ঘটে। | ৩ |
| onstorage | যখন ওয়েব স্টোরেজ এরিয়া আপডেট হয় তখন এই ঘটনা ঘটে। | ৩ |
| [ontoggle](https://www.sattacademy.com/jsref/event_ontoggle.php) | ইউজার যখন <details> এলিমেন্ট ওপেন অথবা ক্লোজ করে তখন এই ঘটনা ঘটে। | ৩ |
| [onwheel](https://www.sattacademy.com/jsref/event_onwheel.php) | যখন এলিমেন্টের উপর মাউস এর হুইল উপরে অথবা নিচে ঘুরানো হয় তখন এই ঘটনা ঘটে। | ৩ |

## টাচ ইভেন্ট

|  |  |  |
| --- | --- | --- |
| **ইভেন্ট** | **বর্ণনা** | **ডোম** |
| ontouchcancel | যখন টাচ বিঘ্নিত হয় তখন এই ঘটনা ঘটে। |  |
| ontouchend | যখন টাচ স্ক্রিন থেকে আঙ্গুল সরিয়ে নেয়া হয় তখন এই ঘটনা ঘটে। |  |
| ontouchmove | যখন স্ক্রিনের উপর আঙ্গুল ড্রাগ করা হয় তখন এই ঘটনা ঘটে। |  |
| ontouchstart | যখন টাচ স্ক্রিনের উপর আঙ্গুল নেয়া হয় তখন এই ঘটনা ঘটে। |  |

## ইভেন্ট অবজেক্ট

### কন্সট্যান্ট

|  |  |  |
| --- | --- | --- |
| **কন্সট্যান্ট** | **বর্ণনা** | **ডোম** |
| CAPTURING\_PHASE | বর্তমান ইভেন্ট ফেজ হচ্ছে ক্যাপচার ফেজ(১) | ১ |
| AT\_TARGET | বর্তমান ইভেন্ট হচ্ছে টার্গেট ফেজ। | ২ |
| BUBBLING\_PHASE | বর্তমান ইভেন্ট ফেজ হচ্ছে বাবলিং ফেজ(৩) | ৩ |

### প্রোপার্টি

|  |  |  |
| --- | --- | --- |
| **প্রোপার্টি** | **বর্ণনা** | **ডোম** |
| [bubbles](https://www.sattacademy.com/jsref/event_bubbles.php) | নির্দিষ্ট ইভেন্ট বাবলিং ইভেন্ট কিনা তা রিটার্ন করে। | ২ |
| [cancelable](https://www.sattacademy.com/jsref/event_cancelable.php) | একটি ইভেন্ট ডিফল্ট অ্যাকশন প্রতিরোধক কিনা তা রিটার্ন করে। | ২ |
| [currentTarget](https://www.sattacademy.com/jsref/event_currenttarget.php) | এলিমেন্টটির কোন ইভেন্ট লিসেনার ইভেন্টিকে ট্রিগার করে তা রিটার্ন করে। | ২ |
| [defaultPrevented](https://www.sattacademy.com/jsref/event_defaultprevented.php) | ইভেন্টের জন্য preventDefault() মেথড কল করা হয়েছে কিনা তা রিটার্ন করে। | ৩ |
| [eventPhase](https://www.sattacademy.com/jsref/event_eventphase.php) | ইভেন্ট ফ্লো এর ফেজ বর্তমানে সংখ্যা গননা করছে কিনা তা রিটার্ন করে। | ২ |
| [isTrusted](https://www.sattacademy.com/jsref/event_istrusted.php) | একটি ইভেন্ট ট্রাস্টেড কিনা তা রিটার্ন করে। | ৩ |
| [target](https://www.sattacademy.com/jsref/event_target.php) | যে এলিমেন্টটি ইভেন্টকে ট্রিগার করে তা রিটার্ন করে। | ২ |
| [timeStamp](https://www.sattacademy.com/jsref/event_timestamp.php) | যখন এলিমেন্টটি তৈরি হয় সেই সময় রিটার্ন করে।(মিলিসেকেন্ডে) | ২ |
| [type](https://www.sattacademy.com/jsref/event_type.php) | ইভেন্টের নাম রিটার্ন করে। | ২ |
| [view](https://www.sattacademy.com/jsref/event_view.php) | যেখানে ইভেন্ট ঘটে উইন্ডো অবজেক্টে একটি রেফারেন্স রিটার্ন করে। | ২ |

### মেথড

|  |  |  |
| --- | --- | --- |
| **মেথড** | **বর্ণনা** | **ডোম** |
| [preventDefault()](https://www.sattacademy.com/jsref/event_preventdefault.php) | যদি ইভেন্টটি বাদ দেয়ার মত হয় তাহলে বাদ দিবে, তার মানে হচ্ছে ডিফল্ট অ্যাকশন ইভেন্ট ঘটবে না। | ২ |
| [stopImmediatePropagation()](https://www.sattacademy.com/jsref/event_stopimmediatepropagation.php) | একই ইভেন্ট হতে অন্যান্য শ্রোতাদের বিরত থাতে বলা হয়। | ৩ |
| stopPropagation() | ইভেন্ট ফ্লো করার সময় একটি ইভেন্টের অধিক বিস্তার থেকে বিরত রাখে। | ২ |

## মাউস ইভেন্ট অবজেক্ট

|  |  |  |
| --- | --- | --- |
| **প্রোপার্টি** | **বর্ণনা** | **ডোম** |
| [altKey](https://www.sattacademy.com/jsref/event_altkey.php) | যখন মাউস ইভেন্ট ট্রিগার করা হয় তখন "ALT" বাটন ক্লিক করা হয়েছে কিনা তা রিটার্ন করে। | ২ |
| [button](https://www.sattacademy.com/jsref/event_button.php) | যখন মাউস ইভেন্ট ট্রিগার করা হয় তখন মাউস বাটন প্রেস হয়েছে কিনা তা রিটার্ন করে। | ২ |
| [buttons](https://www.sattacademy.com/jsref/event_buttons.php) | যখন মাউস ইভেন্ট ট্রিগার করা হয় তখন কোন বাটনটি প্রেস করা হয়েছে তা রিটার্ন করে। | ৩ |
| [clientX](https://www.sattacademy.com/jsref/event_clientx.php) | যখন মাউস ইভেন্ট ট্রিগার করা হয় তখন বর্তমান উইন্ডোর সাপেক্ষে মাউস পয়েন্টারের অনুভুমিক স্থানাঙ্ক রিটার্ন করে। | ২ |
| [clientY](https://www.sattacademy.com/jsref/event_clienty.php) | যখন মাউস ইভেন্ট ট্রিগার করা হয় তখন বর্তমান উইন্ডোর সাপেক্ষে মাউস পয়েন্টারের উলম্ব স্থানাঙ্ক রিটার্ন করে। | ২ |
| [ctrlKey](https://www.sattacademy.com/jsref/event_ctrlkey.php) | যখন মাউস ইভেন্ট ট্রিগার করা হয় তখন "CTRL" বাটন ক্লিক করা হয়েছে কিনা তা রিটার্ন করে। | ২ |
| [detail](https://www.sattacademy.com/jsref/event_detail.php) | কতবার মাউস ক্লিক করা হয়েছে তা রিটার্ন করে। | ২ |
| [metaKey](https://www.sattacademy.com/jsref/event_metakey.php) | যখন মাউস ইভেন্ট ট্রিগার করা হয় তখন "META" বাটন ক্লিক করা হয়েছে কিনা তা রিটার্ন করে। | ২ |
| [pageX](https://www.sattacademy.com/jsref/event_pagex.php) | যখন মাউস ইভেন্ট ট্রিগার করা হয় তখন ডকুমেন্টের সাপেক্ষে মাউস পয়েন্টারের অনুভুমিক স্থানাঙ্ক রিটার্ন করে। |  |
| [pageY](https://www.sattacademy.com/jsref/event_pagey.php) | যখন মাউস ইভেন্ট ট্রিগার করা হয় তখন ডকুমেন্টের সাপেক্ষে মাউস পয়েন্টারের উলম্ব স্থানাঙ্ক রিটার্ন করে। |  |
| [relatedTarget](https://www.sattacademy.com/jsref/event_relatedtarget.php) | মাউস ইভেন্টকে ট্রিগার করে এলিমেন্ট সম্পর্কিত এলিমেন্ট রিটার্ন করে। | ২ |
| [screenX](https://www.sattacademy.com/jsref/event_screenx.php) | যখন মাউস ইভেন্ট ট্রিগার করা হয় তখন স্ক্রিনের সাপেক্ষে মাউস পয়েন্টারের অনুভুমিক স্থানাঙ্ক রিটার্ন করে। | ২ |
| [screenY](https://www.sattacademy.com/jsref/event_screeny.php) | যখন মাউস ইভেন্ট ট্রিগার করা হয় তখন স্ক্রিনের সাপেক্ষে মাউস পয়েন্টারের উলম্ব স্থানাঙ্ক রিটার্ন করে। | ২ |
| [shiftKey](https://www.sattacademy.com/jsref/event_shiftkey.php) | যখন মাউস ইভেন্ট ট্রিগার করা হয় তখন "SHIFT" বাটন ক্লিক করা হয়েছে কিনা তা রিটার্ন করে। | ২ |
| [which](https://www.sattacademy.com/jsref/event_which.php) | যখন মাউস ইভেন্ট ট্রিগার করা হয় তখন কোন বাটনটি প্রেস করা হয়েছে তা রিটার্ন করে। | ২ |

## KeyboardEvent অবজেক্ট

|  |  |  |
| --- | --- | --- |
| **প্রোপার্টি** | **বর্ণনা** | **DOM** |
| [altKey](https://www.sattacademy.com/jsref/event_key_altkey.php) | "ALT" key ইভেন্ট ট্রিগার করা হয় তখন কোন বাটনটি প্রেস করা হয়েছে তা রিটার্ন করে। | ২ |
| [ctrlKey](https://www.sattacademy.com/jsref/event_key_ctrlkey.php) | "CTRL" key ইভেন্ট ট্রিগার করা হয় তখন কোন বাটনটি প্রেস করা হয়েছে তা রিটার্ন করে। | ২ |
| [charCode](https://www.sattacademy.com/jsref/event_key_charcode.php) | key এর ইউনিকোড ক্যারেক্টার কে রিটার্ন করে যেটি onkeypress ইভেন্টকে ট্রিগার করে। | ২ |
| [key](https://www.sattacademy.com/jsref/event_key_key.php) | key ইভেন্ট দ্বারা key এর ভ্যালু রিটার্ন করে। | ৩ |
| [keyCode](https://www.sattacademy.com/jsref/event_key_keycode.php) | key এর ইউনিকোড ক্যারেক্টার কে রিটার্ন করে যেটি onkeypress ইভেন্টকে ট্রিগার করে | ২ |
| [location](https://www.sattacademy.com/jsref/event_key_location.php) | কিবোর্ড অথবা ডিভাইসে key এর লোকেশন রিটার্ন করে। | ৩ |
| [metaKey](https://www.sattacademy.com/jsref/event_key_metakey.php) | যখন key ইভেন্ট ট্রিগার করা হয়,"meta" key প্রেস করা হয়েছে কিনা তা রিটার্ন করে । | ২ |
| [shiftKey](https://www.sattacademy.com/jsref/event_key_shiftkey.php) | যখন key ইভেন্ট ট্রিগার করা হয়, "SHIFT" key প্রেস করা হয়েছে কিনা তা রিটার্ন করে। | ২ |

## HashChangeEvent অবজেক্ট

|  |  |  |
| --- | --- | --- |
| **প্রোপার্টি** | **বর্ণনা** | **ডোম** |
| [newURL](https://www.sattacademy.com/jsref/event_hashchange_newurl.php) | ডকুমেন্টের নতুন URL কে রিটার্ন করে |  |
| [oldURL](https://www.sattacademy.com/jsref/event_hashchange_oldurl.php) | hash পরিবরতন করার পূর্বে ডকুমেন্টের পুরনো URL কে রিটার্ন করে |  |

## PageTransitionEvent অবজেক্ট

|  |  |  |
| --- | --- | --- |
| **প্রোপার্টি** | **বর্ণনা** | **ডোম** |
| persisted | ব্রাউজার দ্বারা ওয়েব পেইজ কেস করে কিনা তা রিটার্ন করে |  |

## FocusEvent অবজেক্ট

|  |  |  |
| --- | --- | --- |
| **প্রোপার্টি** | **বর্ণনা** | **ডোম** |
| relatedTarget | এলিমেন্টের সাথে সম্পৃক্ত এলিমেন্ট কে রিটার্ন করে যেটি ইভেন্টের ট্রিগার | ৩ |

## AnimationEvent অবজেক্ট

|  |  |  |
| --- | --- | --- |
| **প্রোপার্টি** | **বর্ণনা** | **ডোম** |
| animationName | এনিমেশনের নাম রিটার্ন করে |  |
| elapsedTime | একটি এনিমেশনের চলার সময় কে সেকেন্ডে রিটার্ন করে |  |

## TransitionEvent অবজেক্ট

|  |  |  |
| --- | --- | --- |
| **প্রোপার্টি** | **বর্ণনা** | **ডোম** |
| propertyName | ট্রানজিশন সঙ্গে যুক্ত সিএসএস প্রোপার্টির নাম রিটার্ন করে |  |

## WheelEvent অবজেক্ট

|  |  |  |
| --- | --- | --- |
| **প্রোপার্টি** | **বর্ণনা** | **ডোম** |
| deltaX | মাউসের অনুভুমিক স্কলের পরিমান রিটার্ন করে | ৩ |
| deltaY | মাউসের উলম্ব স্কলের পরিমান রিটার্ন করে | ৩ |

## জাভাস্ক্রিপ্ট কি করতে পারে?

ইভেন্ট হ্যান্ডেলার নিয়ন্ত্রণ, যাচাই, ইউজার ইনপুট, ব্যবহারকারীর কোন কাজ এবং ব্রাউজারের ক্রিয়ার ব্যবহার করতে পারেনঃ

* প্রতিবার পেজ লোড হলে কিছু কাজ সম্পাদন হবে
* পেজ বন্ধ করলে কিছু কাজ সম্পাদন হবে
* ব্যবহারকারী একটি বাটনে ক্লিক করলে কিছু কাজ ঘটবে
* ব্যবহারকারী কোন ডাটা প্রদান করবে তখন কন্টেন্টগুলোকে যাচাই করা হবে এবং আরো...

ইভেন্টের কার্য সম্পাদনে জাভাস্ক্রিপ্টে বিভিন্ন ধরনের পদ্ধতি রয়েছেঃ

* এইচটিএমএল ইভেন্ট এট্রিবিউট জাভাস্ক্রিপ্টের কোড সরাসরি এক্সিকিউট করতে পারে
* এইচটিএমএল ইভেন্ট এট্রিবিউট জাভাস্ক্রিপ্টের ফাংশনকে কল করতে পারে
* এইচটিএমএল এলিমেন্টে আপনার নিজস্ব ইভেন্ট হ্যান্ডেলার ফাংশনকে এসাইন করতে পারেন।
* ইভেন্টকে আপনি প্রতিরোধ বা পরিচালিত করা থেকে বিরত রাখতে পারেন এবং আরো...

# জাভাস্ক্রিপ্ট টাইপ কনভার্সন

Number() মেথড নম্বরে, String() মেথড স্ট্রিং-এ, Boolean() মেথড বুলিয়ানে রুপান্তর করে।

## জাভাস্ক্রিপ্ট ডাটা টাইপ

জাভাস্ক্রিপ্টে ৫ ধরনের ডাটা টাইপ রয়েছে যাদের মধ্যে ভ্যালু থাকেঃ

* string
* number
* boolean
* object
* function

অবজেক্ট ৩ ধরনের হয়ঃ

* Object
* Date
* Array

এবং 2 ধরনের ডাটা টাইপ রয়েছে যাদের ভ্যালু নেইঃ

* Null
* Undefined

## typeof অপারেটর

জাভাস্ক্রিপ্ট ভ্যারিয়েবলের ডাটা টাইপ পাওয়ার জন্য **typeof** অপারেটর ব্যবহার করা হয়।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> typeof অপারেটর দ্বারা ভ্যারিয়েবল, অবজেক্ট, ফাংশন, অথবা এক্সপ্রেশন এর টাইপকে বুঝায় ।</h3>

<p id="test"></p>

<script>

document.getElementById("test").innerHTML =

typeof "Tahmid" + "<br>" +

typeof 3.14 + "<br>" +

typeof NaN + "<br>" +

typeof false + "<br>" +

typeof new Date() + "<br>" +

typeof function () {} + "<br>" +

typeof newDay + "<br>" +

typeof [1,2,3,4] + "<br>" +

typeof {name:'আজিজুর', age:34} + "<br>" +

typeof null;

</script>

</body>

</html><

Copy

### ফলাফল

### typeof অপারেটর দ্বারা ভ্যারিয়েবল, অবজেক্ট, ফাংশন, অথবা এক্সপ্রেশন এর টাইপকে বুঝায় ।

string

number

number

boolean

object

function

undefined

object

object

object

লক্ষ্য করুনঃ

* NaN এর ডাটা টাইপ হলো number
* array এর ডাটা টাইপ হলো object
* date এর ডাটা টাইপ হলো object
* null এর ডাটা টাইপ হলো object
* একটি আনডিফাইন্ড ভ্যারিয়েবলের ডাটা টাইপ হলো  **undefined**\*
* ভ্যালু এসাইন করা হয়নি এমন ভ্যারিয়েবলের ডাটা টাইপ হলো  **undefined**\*

একটি জাভাস্ক্রিপ্ট অবজেক্ট অ্যারে অথবা তারিখ কিনা যাচাই করার জন্য আপনি **typeof** অপারেটর ব্যবহার করতে পারবেন না।

## typeof এর ডাটা টাইপ

typeof অপারেটর ভ্যারিয়েবল নয়। ইহা একটি অপারেটর। অপারেটরের ( + - \* /) কোনো ডাটা টাইপ থাকে না।

কিন্তু, typeof অপারেটর অপারেন্ডের টাইপ ধারন করে সবসময় একটি স্ট্রিং রিটার্ন করে।

## constructor প্রোপার্টি

**constructor** প্রোপার্টি সকল জাভাস্ক্রিপ্ট ভ্যারিয়েবলের কন্সট্রাক্টর ফাংশন রিটার্ন করে।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>ভেরিয়েভল অথবা অবজেক্টের জন্য কন্সট্রাক্টর প্রোপার্টি কন্সট্রাক্টর ফাংশনকে রির্টান করে । </h3>

<p id="test"></p>

<script>

document.getElementById("test").innerHTML =

"আজিজুর".constructor + "<br>" +

(3.14).constructor + "<br>" +

false.constructor + "<br>" +

[1,2,3,4].constructor + "<br>" +

{name:'আজিজুর', age:24}.constructor + "<br>" +

new Date().constructor + "<br>" +

function () {}.constructor;

</script>

</body>

</html>

Copy

### ফলাফল

### ভেরিয়েভল অথবা অবজেক্টের জন্য কন্সট্রাক্টর প্রোপার্টি কন্সট্রাক্টর ফাংশনকে রির্টান করে ।

function String() { [native code] }

function Number() { [native code] }

function Boolean() { [native code] }

function Array() { [native code] }

function Object() { [native code] }

function Date() { [native code] }

function Function() { [native code] }

একটি অবজেক্ট অ্যারে(তার মধ্যে "Array" শব্দটি আছে কিনা) কিনা যাচাই করার জন্য আপনি constructor প্রোপার্টি ব্যবহার করতে পারেনঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>এই "হোম মেথড" ফাংশনটি যখন অ্যারেতে ব্যবহৃত হয়,তখন true রিটার্ন করে। </h3>

<p id="test"></p>

<script>

var firstName = ["Tamim", "Tahmid", "Motiur", "Sarif"];

document.getElementById("test").innerHTML = isArray(firstName);

function isArray(x) {

return x.constructor.toString().indexOf("Array") > -1;

}

</script>

</body>

</html>

Copy

### ফলাফল

### এই "হোম মেথড" ফাংশনটি যখন অ্যারেতে ব্যবহৃত হয়,তখন true রিটার্ন করে।

true

একটি অবজেক্ট Date(তার মধ্যে "Date" শব্দটি আছে কিনা) কিনা যাচাই করার জন্য আপনি constructor প্রোপার্টি ব্যবহার করতে পারেনঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>যখন তারিখ ব্যবহার করা হয় ,তখন "home made" ফাংশনটি true রির্টান করে ।</h3>

<p id="test"></p>

<script>

var currentDate = new Date();

document.getElementById("test").innerHTML = isDate(currentDate);

function isDate(currentDate) {

return myDate.constructor.toString().indexOf("Date") > -1;

}

</script>

</body>

</html>

Copy

### ফলাফল

### যখন তারিখ ব্যবহার করা হয় ,তখন "home made" ফাংশনটি true রির্টান করে ।

true

## জাভাস্ক্রিপ্ট টাইপ কনভার্সন

জাভাস্ক্রিপ্ট ভ্যারিয়েবলকে একটি নতুন ভ্যারিয়েবল এবং অন্য ডাটা টাইপে রূপান্তরিত যায়ঃ

* একটি জাভাস্ক্রিপ্ট ফাংশন ব্যবহার করে
* স্বয়ংক্রিয়ভাবে জাভাস্ক্রিপ্টের মাধ্যমে

## নম্বরকে স্ট্রিং-এ রূপান্তর

**String()** গ্লোবাল মেথডটি নম্বরকে স্ট্রিং-এ রূপান্তরিত করে।

এতে যেকোন ধরনের নম্বর, লিটারাল, ভ্যারিয়েবল বা এক্সপ্রেশন ব্যবহার করা যাবেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> String() মেথডটি একটি নম্বরকে স্ট্রিং-এ রূপান্তরিত করতে পারে ।</h3>

<p id="test"></p>

<script>

var x = 192;

document.getElementById("test").innerHTML =

String(x) + "<br>" + // একটি নম্বর ভ্যারিয়েবল x কে স্ট্রিং আকারে রিটার্ন করে

String(192) + "<br>" + // একটি লিটারাল নম্বর 192 কে স্ট্রিং-এ রিটার্ন করে

String(93 + 99); // একটি এক্সপ্রেশনের নম্বরকে স্ট্রিং রিটার্ন করে

</script>

</body>

</html>

Copy

### ফলাফল

### String() মেথডটি একটি নম্বরকে স্ট্রিং-এ রূপান্তরিত করতে পারে ।

192

192

192

Number মেথডের **toString()** একই কাজ করে।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> toString() মেথড একটি সংখ্যাকে স্ট্রিং-এ রুপান্তর করে। </h3>

<p id="test"></p>

<script>

var x = 99;

document.getElementById("test").innerHTML =

x.toString() + "<br>" +

(99).toString() + "<br>" +

(90 + 9).toString();

</script>

</body>

</html>

Copy

### ফলাফল

### toString() মেথড একটি সংখ্যাকে স্ট্রিং-এ রুপান্তর করে।

99

99

99

আমাদের [Number মেথড](https://www.sattacademy.com/js/js_number_methods.php) পরিচ্ছেদে আরো অনেক মেথড পাবেন।

# জাভাস্ক্রিপ্ট নম্বর মেথড

নম্বর মেথড নম্বর নিয়ে কাজ করতে সাহায্য করে।

## নম্বর মেথড এবং প্রোপার্টি

প্রিমিটিভ ভ্যালুর(যেমন ৩.১৪ অথবা ২০১৬) প্রোপার্টি এবং মেথড থাকে না(কারন তারা অবজেক্ট না)।

কিন্তু জাভাস্ক্রিপ্টে, প্রিমিটিভ ভ্যালুর প্রোপার্টি এবং মেথড থাকে, কারন জাভাস্ক্রিপ্ট মেথড এবং প্রোপার্টিগুলোর এক্সিকিউশনের সময় প্রিমিটিভ ভ্যালুকে অবজেক্ট হিসেবে বিবেচনা করে।

## toString() মেথড

**toString()** মেথড নম্বরকে স্ট্রিং আকারে রিটার্ন করে।

সকল নম্বর মেথড যেকোনো ধরনের নম্বরের(লিটারাল, ভ্যাারিয়েবল অথবা এক্সপ্রেশন) ক্ষেত্রে ব্যবহার করা যায়ঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> toString() মেথড একটি সংখ্যাকে স্ট্রিং-এ রুপান্তর করে। </h4>

<p id="test"></p>

<script>

var a = 198;

document.getElementById("test").innerHTML =

a.toString() + "<br>" +

(198).toString() + "<br>" +

(100 + 98).toString();

</script>

</body>

</html>

Copy

### ফলাফল

#### toString() মেথড একটি সংখ্যাকে স্ট্রিং-এ রুপান্তর করে।

198  
198  
198

## toExponential() মেথড

**toExponential()** মেথড স্ট্রিং আকারে একটি সংখ্যাকে পূর্ণ নম্বরে পরিনত করে এবং একে বীজগানিতিক নোটেশন আকারে লিখে রিটার্ন করে।

দশমিক নম্বরের পরে কয়টি ক্যারেক্টার থাকবে একটি প্যারামিটার নম্বরের মাধ্যমে নির্দিষ্ট করে দেওয়া হয়ঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> toExponential() মেথড সংখ্যা গোলাকার এবং সূচকীয় স্বরলিপি ব্যবহার করে একটি স্ট্রিং রিটার্ন করে। </h4>

<p>একটি ঐচ্ছিক প্যারামিটার দশমিক পয়েন্টে পিছনের ডিজিটের সংখ্যা নির্ধারণ করে। </p>

<p id="test"></p>

<script>

var a = 7.567;

document.getElementById("test").innerHTML =

a.toExponential() + "<br>" +

a.toExponential(2) + "<br>" +

a.toExponential(4) + "<br>" +

a.toExponential(6);

</script>

</body>

</html>

Copy

### ফলাফল

#### toExponential() মেথড সংখ্যা গোলাকার এবং সূচকীয় স্বরলিপি ব্যবহার করে একটি স্ট্রিং রিটার্ন করে।

একটি ঐচ্ছিক প্যারামিটার দশমিক পয়েন্টে পিছনের ডিজিটের সংখ্যা নির্ধারণ করে।

7.567e+0  
7.57e+0  
7.5670e+0  
7.567000e+0

## toFixed() মেথড

**toFixed()** মেথড একটি সংখ্যাকে নির্দিষ্ট সংখ্যক দশমিক নম্বরে লিখে একটি স্ট্রিং রিটার্ন করেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>toFixed () মেথড ডিজিটের একটি প্রদত্ত সংখ্যার একটি নম্বর চক্রকে বুঝায়। </h4>

<p>টাকার বিষয় নিয়ে কাজ করার জন্য, toFixed (2) নির্ভুল। </p>

<p id="test"></p>

<script>

var a = 7.567;

document.getElementById("test").innerHTML =

a.toFixed(0) + "<br>" +

a.toFixed(2) + "<br>" +

a.toFixed(4) + "<br>" +

a.toFixed(6);

</script>

</body>

</html>

Copy

### ফলাফল

#### toFixed () মেথড ডিজিটের একটি প্রদত্ত সংখ্যার একটি নম্বর চক্রকে বুঝায়।

টাকার বিষয় নিয়ে কাজ করার জন্য, toFixed (2) নির্ভুল।

8  
7.57  
7.5670  
7.567000

toFixed(2) মেথড টাকা সংক্রান্ত বিষয় নিয়ে কাজ করার জন্য উপযুক্ত।

## toPrecision() মেথড

**toPrecision()** মেথড একটি নির্দিষ্ট দৈর্ঘ্যের সংখ্যাকে স্ট্রিং আকারে রিটার্ন করেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>যখন একটি নম্বরকে নির্দিষ্ট দৈর্ঘ্যে লিখা হয় তখন toPrecision() মেথড একটি স্ট্রিং কে রিটার্ণ করেঃ</h4>

<p id="test"></p>

<script>

var a = 7.567;

document.getElementById("test").innerHTML =

a.toPrecision() + "<br>" +

a.toPrecision(2) + "<br>" +

a.toPrecision(4) + "<br>" +

a.toPrecision(6);

</script>

</body>

</html>

Copy

### ফলাফল

#### যখন একটি নম্বরকে নির্দিষ্ট দৈর্ঘ্যে লিখা হয় তখন toPrecision() মেথড একটি স্ট্রিং কে রিটার্ণ করেঃ

7.567  
7.6  
7.567  
7.56700

## valueOf() মেথড

**valueOf()** মেথড একটি নম্বরকে নম্বর হিসেবে রিটার্ন করে।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var a = 198;

document.getElementById("test").innerHTML =

a.valueOf() + "<br>" +

(198).valueOf() + "<br>" +

(100 + 98).valueOf();

</script>

</body>

</html>

Copy

### ফলাফল

### 198 198 198

জাভাস্ক্রিপ্টে, একটি নম্বর প্রিমিটিভ ভ্যালু(typeof = number) অথবা একটি অবজেক্ট(typeof = object) হতে পারে।

জাভাস্ক্রিপ্টে ইন্টার্নাল্লি নম্বর অবজেক্টকে প্রিমিটিভ ভ্যালুতে রুপান্তর করার জন্য valueOf() মেথড ব্যবহার করা হয়েছে। আপনার কোডে ইহা ব্যবহারের প্রয়োজন নেই।

সকল জাভাস্ক্রিপ্ট ডাটা টাইপের একটি valueOf() এবং একটি toString() মেথড আছে।

## ভ্যারিয়েবলকে নম্বরে রুপান্তর

ভ্যারিয়েবলকে নম্বরে রুপান্তর করার জন্য ৩ ধরনের জাভাস্ক্রিপ্ট মেথড ব্যবহার করা হয়ঃ

* Number() মেথড
* parseInt() মেথড
* parseFloat() মেথড

এই মেথডগুলো **নম্বর** মেথড নয়, কিন্তু **গ্লোবাল** জাভাস্ক্রিপ্ট মেথড।

## গ্লোবাল মেথড

জাভাস্ক্রিপ্ট গ্লোবাল মেথডগুলো সকল জাভাস্ক্রিপ্ট ডাটা টাইপে ব্যবহার করা যায়।

যখন নম্বর নিয়ে কাজ করা হবে তখন এইগুলোই নম্বর সম্পর্কিত মেথডঃ

|  |  |
| --- | --- |
| **মেথড** | **বর্ণনা** |
| Number() | আর্গুমেন্টকে রুপান্তর করে একটি নম্বর রিটার্ন করে |
| parseFloat() | আর্গুমেন্ট পার্স করে এবং দশমিকযুক্ত নম্বর রিটার্ন করে |
| parseInt() | আর্গুমেন্ট পার্স করে এবং পূর্ণ নম্বর রিটার্ন করে |

## Number() মেথড

**Number()** মেথড জাভাস্ক্রিপ্ট ভ্যারিয়েবলকে নম্বরে রুপান্তরের ক্ষেত্রে ব্যবহার করা যেতে পারেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>Number() মেথড জাভাস্ক্রিপ্ট ভেরিয়েবল গুলোকে নম্বরে রুপান্তরের ক্ষেত্রে ব্যবহার করা যেতে পারেঃ</h4>

<p id="test"></p>

<script>

document.getElementById("test").innerHTML =

Number(true) + "<br>" +

Number(false) + "<br>" +

Number(new Date()) + "<br>" +

Number(" 13") + "<br>" +

Number("13 ") + "<br>" +

Number("13 5");

</script>

</body>

</html>

Copy

### ফলাফল

#### Number() মেথড জাভাস্ক্রিপ্ট ভেরিয়েবল গুলোকে নম্বরে রুপান্তরের ক্ষেত্রে ব্যবহার করা যেতে পারেঃ

1  
0  
1532603889982  
13  
13  
NaN

Date() মেথড Number() মেথডে ব্যবহার করলে ১.১.১৯৭০ সাল থেকে মিলিসেকেন্ডকে নম্বরে রিটার্ন করে।

## parseInt() মেথড

**parseInt()** মেথড একটি স্ট্রিংকে পার্স করে এবং একটি সম্পূর্ণ নম্বর রিটার্ন করে। এক্ষেত্রে স্পেস দেওয়া যায়। স্পেস দিলে শুধুমাত্র প্রথম নম্বরকেই রিটার্ন করেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>parseInt() মেথড একটি স্ট্রিংকে parse করে এবং একটি পুরো নম্বরকে রিটার্ণ করে। </h4>

<p id="test"></p>

<script>

document.getElementById("test").innerHTML =

parseInt("13") + "<br>" +

parseInt("13.33") + "<br>" +

parseInt("13 6") + "<br>" +

parseInt("13 scores") + "<br>" +

parseInt("scores 13");

</script>

</body>

</html>

Copy

### ফলাফল

#### parseInt() মেথড একটি স্ট্রিংকে parse করে এবং একটি পুরো নম্বরকে রিটার্ণ করে।

13  
13  
13  
13  
NaN

যদি নম্বরে পরিবর্তিত না হয়, তাহলে NaN(Not a Number) রিটার্ন করবে।

## parseFloat() মেথড

**parseFloat()** মেথড স্ট্রিংকে পার্স করে এবং একটি নম্বর রিটার্ন করে। এক্ষেত্রে স্পেস দেওয়া যায়। স্পেস দিলে শুধুমাত্র প্রথম নম্বরকেই রিটার্ন করেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>parseFloat() মেথড একটি স্ট্রিংকে parse করে এবং একটি নম্বরকে রিটার্ণ করে। </h4>

<p id="test"></p>

<script>

document.getElementById("test").innerHTML =

parseFloat("13") + "<br>" +

parseFloat("13.33") + "<br>" +

parseFloat("13 6") + "<br>" +

parseFloat("13 scores") + "<br>" +

parseFloat("scores 13");

</script>

</body>

</html>

Copy

### ফলাফল

#### parseFloat() মেথড একটি স্ট্রিংকে parse করে এবং একটি নম্বরকে রিটার্ণ করে।

13  
13.33  
13  
13  
NaN

যদি নম্বরে পরিবর্তিত না হয়, তাহলে NaN(Not a Number) রিটার্ন করবে।

## নম্বর প্রোপার্টি

|  |  |
| --- | --- |
| **প্রোপার্টি** | **বর্ননা** |
| MAX\_VALUE | জাভাস্ক্রিপ্টে সম্ভাব্য বড় নম্বর রিটার্ন করে |
| MIN\_VALUE | জাভাস্ক্রিপ্টে সম্ভাব্য ছোট নম্বর রিটার্ন করে |
| NEGATIVE\_INFINITY | ঋণাত্বক অসীম মান প্রদর্শন করে |
| NaN | "Not-a-Number" ভ্যালু প্রদর্শন করে |
| POSITIVE\_INFINITY | ধনাত্বক অসীম মান প্রদর্শন করে |

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

document.getElementById("test").innerHTML = Number.MAX\_VALUE;

</script>

</body>

</html>

Copy

### ফলাফল

1.7976931348623157e+308

নম্বর প্রোপার্টিগুলো জাভাস্ক্রিপ্ট নম্বর অবজেক্টের অধীনস্থ যাকে **নম্বর** হিসেবে ডাকা হয়।

এই সকল প্রোপার্টি শুধুমাত্র **Number.MAX\_VALUE** হিসেবে এক্সেস করা যাবে।

ভ্যারিয়েবল, এক্সপ্রেশন অথবা কোন ভ্যালুতে .MAX\_VALUE প্রোপার্টি ব্যবহার করলে undefined রিটার্ন করবেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var x = 6;

document.getElementById("test").innerHTML = x.MAX\_VALUE;

</script>

</body>

</html>

Copy

### ফলাফল

undefined

## জাভাস্ক্রিপ্ট নম্বর রেফারেন্স পড়ুন

সম্পূর্ণ রেফারেন্সের জন্য আমাদের [জাভাস্ক্রিপ্ট নম্বর রেফারেন্স](https://www.sattacademy.com/jsref/jsref_obj_number.php) দেখুন।

রেফারেন্সে সকল নম্বর প্রোপার্টি এবং মেথডের বর্ণনা দেওয়া হয়েছে।

## জাভাস্ক্রিপ্ট নাম্বার

জাভাস্ক্রিপ্টে দশমিক অথবা দশমিক ছাড়া লেখা যায়ঃ

var a = 2.5;

var b = 2;

Copy

অতিরিক্ত বড় অথবা অতিরিক্ত ছোট নাম্বারগুলোকে বৈজ্ঞানিক সূচক আকারে লিখা যায়ঃ

var a = 25e6;

var b = 25e-6;

Copy

জাভাস্ক্রিপ্ট নাম্বার সম্পর্কে আরো জানতে আমাদের [জাভাস্ক্রিপ্ট নাম্বার টিউটোরিয়ালে](https://www.sattacademy.com/js/js_numbers.php) পড়ুন।

## নাম্বার(Number) প্রোপার্টি

|  |  |
| --- | --- |
| **প্রোপার্টি** | **বর্ণনা** |
| constructor | যে ফাংশনটি জাভাস্ক্রিপ্ট নাম্বার প্রোটোটাইপ তৈরি করে সেটি রিটার্ন করে। |
| MAX\_VALUE | জাভাস্ক্রিপ্টে সম্ভাব্য সর্বোচ্চ নাম্বার রিটার্ন করে। |
| MIN\_VALUE | জাভাস্ক্রিপ্টে সম্ভাব্য সর্বনিম্ন নাম্বার রিটার্ন করে। |
| NEGATIVE\_INFINITY | নেগেটিভ অসীম ভ্যালু প্রকাশ করে। |
| NaN | নাম্বার(Not-a-Number) নয় ভ্যালু প্রকাশ করে। |
| POSITIVE\_INFINITY | পজিটিভ অসীম ভ্যালু প্রকাশ করে। |
| prototype | একটি অবজেক্টে প্রোপার্টি এবং মেথড যোগ করে। |

## নাম্বার(Number) মেথড

|  |  |
| --- | --- |
| **মেথড** | **বর্ণনা** |
| isFinite() | সসীম মান কিনা তা চেক করে। |
| isInteger() | পূর্ণ সংখ্যার ভ্যালু কিনা তা চেক করে। |
| isNaN() | ভ্যালুটি নাম্বার নয় কিনা তা চেক করে। |
| isSafeInteger() | ভ্যালুটি নিরাপদ পূর্ণ সংখ্যা কিনা তা চেক করে। |
| toExponential(x) | একটি নাম্বারকে বীজগণিতীয় রাশির প্রতীকে পরিনত করে। |
| toFixed(x) | একটি নাম্বারকে দশমিক সংখ্যার পরে x সংখ্যক ডিজিটসহ প্রদর্শন করে। |
| toPrecision(x) | একটি নাম্বারকে x সংখ্যক ডিজিটসহ প্রদর্শন করে। |
| toString() | একটি নাম্বারকে স্ট্রিং-এ পরিনত করে। |
| valueOf() | একটি নাম্বারের মৌলিক মান রিটার্ন করে। |

সকল নাম্বার মেথড একটি নতুন ভ্যালু রিটার্ন করে। কিন্তু অরিজিনাল ভ্যালুকে পরিবর্তন করে না।

## বুলিয়ানকে স্ট্রিং-এ রূপান্তর

**String()** গ্লোবাল মেথডটি বুলিয়ানকে স্ট্রিং-এ রুপান্তরিত করে।

String(false) // "false" রিটার্ন করে

String(true) // "true" রিটার্ন করে

Copy

**toString()** বুলিয়ান মেথডও একই কাজ করে।

false.toString() // "false" রিটার্ন করে

true.toString() // "true" রিটার্ন করে

Copy

## তারিখকে স্ট্রিং-এ রূপান্তর

**String()** গ্লোবাল মেথডটি তারিখকে স্ট্রিং-এ রুপান্তর করে।

String(Date()) // Wed Mar 22 2017 16:26:40 GMT+0600 (Bangladesh Standard Time) রিটার্ন করে

Copy

**toString()** Date মেথডও একই কাজ করে।

Date().toString() // Wed Mar 22 2017 16:26:40 GMT+0600 (Bangladesh Standard Time) রিটার্ন করে

Copy

আমাদের [Date মেথড](https://www.sattacademy.com/js/js_date_methods.php) পরিচ্ছেদে আরো অনেক মেথড পাবেন।

## স্ট্রিংকে নম্বরে রূপান্তর

**Number()** গ্লোবাল মেথডটি নম্বরকে স্ট্রিং-এ রুপান্তর করে।

স্ট্রিং নম্বর("3.14") নম্বরে(3.14) রুপান্তর হয়।

এম্পটি স্ট্রিং 0 তে রুপান্তরিত হয়।

অন্য সবকিছু NaN হবে।

Number("") // 0 রিটার্ন করে

Number("99 88") // NaN রিটার্ন করে

Number("2.54") // 3.14 রিটার্ন করে

Number(" ") // 0 রিটার্ন করে

Copy

আমাদের [Number মেথড](https://www.sattacademy.com/js/js_number_methods.php) পরিচ্ছেদে আপনি আরো মেথড পাবেন।

## ইউনারী + অপারেটর

**ইউনারী + অপারেটর** একটি ভ্যারিয়েবলকে নম্বরে রুপান্তর করতে ব্যবহার করা যাবেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> typeof অপারেটর দ্বারা ভ্যারিয়েবল অথবা এক্সপ্রেশনকে রির্টান করে । </h3>

<button onclick="myFunc()"> চেষ্টা করি </button>

<p id="test"></p>

<script>

function myFunc() {

var b = "9";

var a = + b;

document.getElementById("test").innerHTML =

typeof b + "<br>" + typeof a;

}

</script>

</body>

</html>

Copy

### ফলাফল

### typeof অপারেটর দ্বারা ভ্যারিয়েবল অথবা এক্সপ্রেশনকে রির্টান করে ।

string  
number

যদি ভ্যারিয়েবলকে রুপান্তর করা না যায় তারপরও এর টাইপ number এ রুপান্তরিত হয়, যার ভ্যালু NaN হবেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>typeof অপারেটর দ্বারা ভ্যারিয়েবল অথবা এক্সপ্রেশনকে রির্টান করে । </h3>

<button onclick="myFunc()"> চেষ্টা করি </button>

<p id="test"></p>

<script>

function myFunc() {

var b = "Tamim";

var a = + b;

document.getElementById("test").innerHTML =

typeof a + "<br>" + a;

}

</script>

</body>

</html>

Copy

### ফলাফল

### typeof অপারেটর দ্বারা ভ্যারিয়েবল অথবা এক্সপ্রেশনকে রির্টান করে ।

number  
NaN

## বুলিয়ানকে নম্বরে রূপান্তর

**Number()** গ্লোবাল মেথডটি বুলিয়ানকে নম্বরে রুপান্তর করে।

Number(false) // 0 রিটার্ন করে

Number(true) // 1 রিটার্ন করে

Copy

## তারিখকে নম্বরে রূপান্তর

**Number()** গ্লোবাল মেথডটি তারিখকে নম্বরে রুপান্তর করেঃ

### উদাহরণ

a = new Date();

Number(a) // রিটার্ন করে 1490189529374

Copy

**getTime()** date মেথডও একই কাজ করে।

a = new Date();

a.getTime() // 1490189529374 রিটার্ন করে

Copy

## স্বয়ংক্রিয় টাইপ কনভার্সন

যখন জাভাস্ক্রিপ্ট একটি "ভুল" ডাটা টাইপ নিয়ে কাজ করে, তখন এটি "সঠিক" টাইপের ভ্যালুতে রূপান্তর করতে চেষ্টা করে।

ফলাফল এমন হতে পারেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var x = [];

document.getElementById("test").innerHTML =

("3" + 1) + "<br>" + // 31 রিটার্ন করে,কারন 1 পরিবর্তিত হয়ে "1" হয়েছে

("3" - 1) + "<br>" + // 2 রিটার্ন করে,কারন "3" পরিবর্তিত হয়ে 3 হয়েছে

("3" \* "1") + "<br>" + // 3 রিটার্ন করে,কারন "3" এবং "1" পরিবর্তিত হয়ে 3 এবং 1 হয়েছে

(3 + null) + "<br>" + // 3 রিটার্ন করে,কারন null পরিবর্তিত হয়ে 0 হয়েছে

("3" + null) + "<br>" + // "3null" রিটার্ন করে,কারন null পরিবর্তিত হয়ে "null" হয়েছে

("3" / "1") + "<br>"

</script>

</body>

</html>

Copy

### ফলাফল

31

2

3

3

3null

3

## স্বয়ংক্রিয় স্ট্রিং কনভার্সন

আপনি একটি অবজেক্ট অথবা ভ্যারিয়েবলের আউটপুট নিতে চাইলে জাভাস্ক্রিপ্ট স্বয়ংক্রিয়ভাবে ভ্যারিয়েবলের toString() ফাংশনকে কল করেঃ

document.getElementById("test").innerHTML = myVar;

// if myVar = {name:"Azizur"} // toString "[object Object]" এ রুপান্তর করে

// if myVar = [1,2,3,4] // toString "1,2,3,4" এ রুপান্তর করে

// if myVar = new Date() // toString "Wed Mar 22 2017 19:46:30 GMT+0600" এ রুপান্তর করে

Copy

নম্বর এবং বুলিয়ানের ক্ষেত্রেও একই রকম ঘটেঃ

// if myVar = 123 // toString "123" এ রুপান্তর করে

// if myVar = true // toString "true" এ রুপান্তর করে

// if myVar = false // toString "false" এ রুপান্তর করে

## জাভাস্ক্রিপ্ট টাইপ কনভার্সন টেবিল

নিম্নোক্ত টেবিলে জাভাস্ক্রিপ্টের বিভিন্ন ভ্যালুকে নম্বর, স্ট্রিং, এবং বুলিয়ানে রুপান্তর করে দেখানো হয়েছেঃ

|  |  |  |  |
| --- | --- | --- | --- |
| **প্রকৃত মান** | **নম্বরে রুপান্তর** | **স্ট্রিং-এ রুপান্তর** | **বুলিয়ানে রুপান্তর** |
| false | 0 | "false" | false |
| true | 1 | "true" | true |
| 0 | 0 | "0" | false |
| 1 | 1 | "1" | true |
| "0" | 0 | "0" | true |
| "1" | 1 | "1" | true |
| NaN | NaN | "NaN" | false |
| Infinity | Infinity | "Infinity" | true |
| -Infinity | -Infinity | "-Infinity" | true |
| "" | 0 | "" | false |
| "20" | 20 | "20" | true |
| "twenty" | NaN | "twenty" | true |
| [ ] | 0 | "" | true |
| [20] | 20 | "20" | true |
| [10,20] | NaN | "10,20" | true |
| ["twenty"] | NaN | "twenty" | true |
| ["ten","twenty"] | NaN | "ten,twenty" | true |
| function(){} | NaN | "function(){}" | true |
| { } | NaN | "[object Object]" | true |
| null | 0 | "null" | false |
| undefined | NaN | "undefined" | false |

কোটেশনের ভ্যালুগুলো স্ট্রিং ভ্যালু নির্দেশ করে।

লাল রংয়ের ভ্যালু দ্বারা বুঝানো হয়েছে প্রোগ্রামাররা এই ভ্যালু প্রত্যাশা করে না।

# জাভাস্ক্রিপ্ট রেগুলার এক্সপ্রেশন

রেগুলার এক্সপ্রেশন হচ্ছে কিছু ক্যারেক্টারের ধারা যা একটি সার্চ প্যাটার্ন তৈরি করে।

সার্চ প্যাটার্নটি টেক্সট খোজার জন্য এবং টেক্সট প্রতিস্থাপনের জন্য ব্যবহার করা যায়।

## রেগুলার এক্সপ্রেশন বলতে কী বোঝায়?

রেগুলার এক্সপ্রেশন হচ্ছে কিছু ক্যারেক্টারের ধারা যা একটি সার্চ প্যাটার্ন তৈরি করে।

টেক্সটের মধ্যে ডাটা খোজার সময় আপনি এই সার্চ প্যাটার্নের মাধ্যমে বলে দিতে পারেন কি পেতে চাইছেন।

রেগুলার এক্সপ্রেশন একটি ক্যারেক্টার অথবা আরো জটিল প্যাটার্নের হতে পারে।

## গঠনপ্রণালী

/pattern/modifiers;

Copy

var patt = /sattacademy/i;

Copy

উদাহরণের ব্যাখ্যাঃ

**/sattacademy/i**  একটি রেগুলার এক্সপ্রেশন।

**sattacademy**  একটি প্যাটার্ন।(একটি অনুসন্ধান ব্যবহার করা হবে)

**i**  একটি মডিফায়ার(যা সার্চকে কেস ইনসেন্সিটিভ হবে নির্দেশ করে)।

## স্ট্রিং মেথডের ব্যবহার

জাভাস্ক্রিপ্টে রেগুলার এক্সপ্রেশন দুটি স্ট্রিং মেথড দ্বারা ব্যবহার করা হয়ঃ search() এবং replace() মেথড।

**search()** মেথড মিল খোজার জন্য এক্সপ্রেশন ব্যবহার করে এবং এর অবস্থান রিটার্ন করে।

**replace()** মেথড পরিবর্তিত স্ট্রিংকে রিটার্ন করে।

## search() মেথডে রেগুলার এক্সপ্রেশনের ব্যবহার

### উদাহরণ

একটি স্ট্রিং-এর মধ্যে "sattacademy" অনুসন্ধান করতে একটি রেগুলার এক্সপ্রেশন ব্যবহার করিঃ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>স্যাট একডেমী এর জন্য একটি স্টিং অনুসন্ধান করে এবং কাঙ্খিত অবস্থান প্রদর্শন করেঃ</h3>

<button onclick="myFunc()"> চেষ্টা করি </button>

<p id="test"></p>

<script>

function myFunc() {

var text = "Visit sattacademy!";

var n = text.search(/sattacademy/i);

document.getElementById("test").innerHTML = n;

}

</script>

</body>

</html>

Copy

### ফলাফল

### স্যাট একডেমী এর জন্য একটি স্টিং অনুসন্ধান করে এবং কাঙ্খিত অবস্থান প্রদর্শন করেঃ

6

search() মেথডে স্ট্রিং-এর ব্যবহার

search মেথড আর্গুমেন্ট হিসাবে একটি স্ট্রিং গ্রহণ করে। স্ট্রিং আর্গুমেন্ট একটি রেগুলার এক্সপ্রেশনে রূপান্তরিত হবেঃ

### উদাহরণ

একটি স্ট্রিং-এর মধ্যে "SATTacademy" অনুসন্ধান করতে একটি স্ট্রিং ব্যবহার করিঃ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> "sattacademy" এর জন্য স্ট্রিং অনুসন্ধান করবে এবং পজিশণ প্রদর্শণ করবেঃ </h3>

<button onclick="myFunc()">ক্লিক</button>

<p id="test"></p>

<script>

function myFunc() {

var text = "Visit sattacademy!";

var n = text.search("sattacademy");

document.getElementById("test").innerHTML = n;

}

</script>

</body>

</html>

Copy

### ফলাফল

### "sattacademy" এর জন্য স্ট্রিং অনুসন্ধান করবে এবং পজিশণ প্রদর্শণ করবেঃ

6

## replace() মেথডে রেগুলার এক্সপ্রেশনের ব্যবহার

### উদাহরণ

একটি স্ট্রিং-এর মধ্যে codeacademy এর স্থানে SATTacademy প্রতিস্থাপন করতে রেগুলার এক্সপ্রেশনের ব্যবহারঃ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> নিচের প্যারাগ্রাফে codeacademy এর স্থানেে sattacademy প্রতিস্থাপন করুনঃ </h3>

<button onclick="myFunc()"> চেষ্টা করি </button>

<p id="test"> ঘুরে আসুন codeacademy!</p>

<script>

function myFunc() {

var text= document.getElementById("txt").innerHTML;

var n = text.replace(/codeacademy/i,"sattacademy");

document.getElementById("txt").innerHTML = n;

}

</script>

</body>

</html>

### ফলাফল

### নিচের প্যারাগ্রাফে codeacademy এর স্থানেে sattacademy প্রতিস্থাপন করুনঃ

## ঘুরে আসুন sattacademy!

## replace() মেথডে স্ট্রিং-এর ব্যবহার

replace() মেথড আর্গুমেন্ট হিসেবে একটি স্ট্রিংও গ্রহণ করেঃ

### উদাহরণ

replace() মেথড আর্গুমেন্ট হিসেবে একটি স্ট্রিংও গ্রহণ করেঃ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> নিচের প্যারাগ্রাফে codeacademy এর স্থানেে sattacademy প্রতিস্থাপন করুনঃ </h3>

<button onclick="myFunc()"> চেষ্টা করি </button>

<p id="test"> ঘুরে আসুন codeacademy!</p>

<script>

function myFunc() {

var text= document.getElementById("test").innerHTML;

var n = text.replace("codeacademy","sattacademy");

document.getElementById("test").innerHTML = n;

}

</script>

</body>

</html>

### ফলাফল

### নিচের প্যারাগ্রাফে codeacademy এর স্থানেে sattacademy প্রতিস্থাপন করুনঃ

ঘুরে আসুন codeacademy!

## লক্ষণীয় বিষয়

উপরের মেথডগুলোতে আর্গুমেন্ট হিসেবে রেগুলার এক্সপ্রেশন ব্যবহার করা যায়।  
রেগুলার এক্সপ্রেশনের ব্যবহার আপনার অনুসন্ধান অনেক বেশি শক্তিশালী করতে পারে।

## রেগুলার এক্সপ্রেশন মডিফায়ার

|  |  |
| --- | --- |
| **মডিফায়ার** | **বর্ণনা** |
| i | অনুসন্ধান কেস-ইনসেন্সিটিভ হবে |
| g | সম্পূর্ন মিলকে খুজে বের করবে |
| m | একাধিক লাইনের মিল খুঁজে বের করে |

## রেগুলার এক্সপ্রেশন প্যাটার্ন

|  |  |
| --- | --- |
| **এক্সপ্রেশন** | **বর্ণনা** |
| [abc] | ব্র্যাকেটের মাঝের যেকোনো অক্ষর খুজে বের করে |
| [0-9] | ব্র্যাকেটের মাঝের যেকোনো সংখ্যা খুজে বের করে |
| (x|y) | | দ্বারা বিভক্ত দুইটির যেকোন একটি খুজে বের করে |

মেটা-ক্যারেক্টার বিশেষ অর্থ সম্বলিত কিছু ক্যারেক্টারঃ

|  |  |
| --- | --- |
| **মেটা-ক্যারেক্টার** | **বর্ণনা** |
| \d | একটি সংখ্যা খুজে বের করে |
| \s | একটি সাদা স্পেস অক্ষর খুজে বের করে |
| \b | একটি শব্দের শুরু অথবা শেষে মিল খুজে বের করে |
| \uxxxx | হেক্সাডেসিমেল xxxx নম্বর দ্বারা নির্দেশিত ইউনিকোড ক্যারেক্টার খুজে বের করে |

কোয়ান্টিফায়ার বিদ্যমানতা বুঝায়ঃ

|  |  |
| --- | --- |
| **কোয়ান্টিফায়ার** | **বর্ণনা** |
| n+ | যেসকল স্ট্রিং-এ অন্তত একটি n রয়েছে তাদেরকে খুজে বের করে |
| n\* | যেসকল স্ট্রিং-এ শূন্য অথবা একের অধিক n রয়েছে তাদের খুজে বের করে |
| n? | যেসকল স্ট্রিং-এ শূন্য অথবা একটি n রয়েছে তাদের খুজে বের করে |

## রেগুলার এক্সপ্রেশন অবজেক্ট ব্যবহার

জাভাস্ক্রিপ্টে রেগুলার এক্সপ্রেশন অবজেক্ট হলো পূর্বনির্ধারিত প্রোপার্টি ও মেথডসহ একটি অবজেক্ট।

## test() মেথডের ব্যবহার

test() মেথড হচ্ছে রেগুলার এক্সপ্রেশনের একটি মেথড।

ইহা একটি স্ট্রিং-এর মধ্যে একটি নমুনা/প্যাটার্ন খোজে,যদি খুজে পায় তাহলে true রিটার্ন করে,না পেলে false রিটার্ন করে।

নিম্নলিখিত উদাহরণটিতে একটি স্ট্রিং-এ "a" অক্ষরটি খুজে বের করবেঃ

var patt = /a/;<br>

patt.test("Be Happy!");

Copy

যেহেতু উপরের স্ট্রিং-এ "e" রয়েছে, সেহেতু ফলাফল হবেঃ

### ফলাফল

  true

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>পরবর্তী প্যারাগ্রাফে "a" অক্ষরটি অনুসন্ধান করি :</h3>

<p id="para1"> Be Happy! "</p>

<button onclick="myFunc()"> চেষ্টা করি </button>

<p id="test"></p>

<script>

function myFunc() {

text = document.getElementById("para1").innerHTML;

document.getElementById("test").innerHTML = /a/.test(text);

}

</script>

</body>

</html>

Copy

### ফলাফল

### পরবর্তী প্যারাগ্রাফে "a" অক্ষরটি অনুসন্ধান করি :

Be Happy! "

true

## exec() ব্যবহার

exec() মেথড একটি রেগুলার এক্সপ্রেশন মেথড।

ইহা একটি স্ট্রিং-এর মধ্যে নির্দিষ্টা নমুনা/প্যাটার্ন খোজে,যদি খুজে পায় তাহলে ঐ টেক্সটটি রিটার্ন করবে। যদি না পাওয়া যায় তবে null রিটার্ন করবে।

নিম্নলিখিত উদাহরণটিতে একটি স্ট্রিং-এ "a" অক্ষরটি খুজে বের করবেঃ

var patt = /a/;<br>

patt.exec("Be Happy!");

Copy

যেহেতু উপরের স্ট্রিং-এ "a" রয়েছে, সেহেতু ফলাফল হবেঃ

### ফলাফল

  a

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>পরবর্তী প্যারাগ্রাফে "a" অক্ষরটি অনুসন্ধান করি :<h3p>

<p id="para1">Be Happy!</p>

<button onclick="myFunc()">চেষ্টা করি </button>

<p id="test"></p>

<script>

function myFunc() {

text = document.getElementById("para1").innerHTML;

document.getElementById("test").innerHTML = /a/.exec(text);

}

</script>

</body>

</html>

Copy

### ফলাফল

পরবর্তী প্যারাগ্রাফে "a" অক্ষরটি অনুসন্ধান করি :

Be Happy! "

### a

### সম্পূর্ণ রেগুলার এক্সপ্রেশন রেফারেন্স

সম্পূর্ণ রেফারেন্সের জন্য আমাদের [জাভাস্ক্রিপ্ট রেগুলার এক্সপ্রেশন রেফারেন্স](https://www.sattacademy.com/jsref/jsref_obj_regexp.php) পড়ুন। রেফারেন্সে সকল রেগুলার এক্সপ্রেশন প্রোপার্টি মেথড এর উদাহরণ এবং বর্ণনা রয়েছে।

# জাভাস্ক্রিপ্ট Error

**try** স্টেটমেন্ট এঁররের জন্য একটি কোড ব্লককে টেস্ট করতে সাহায্য করে

**catch** স্টেটমেন্ট আপনাকে এঁরর নিয়ন্ত্রণ করতে সাহায্য করে

**throw** স্টেটমেন্ট আপনাকে কাস্টম এঁরর তৈরি করতে সাহায্য করে

**finally** স্টেটমেন্ট আপনাকে try এবং catch-এর ফলাফল অনুযায়ী কোড এক্সিকিউট করতে দেয়

## এঁরর!

জাভস্ক্রিপ্ট কোড এক্সিকিউশনের সময় বিভিন্ন ধরনের এঁরর ঘটে।

প্রোগ্রামার দ্বারা কোড এঁরর, ভুল ইনপুট অথবা অন্যান্য যেকোন কারণে এঁরর ঘটতে পারে।

### উদাহরণ

এই উদাহরণে আমরা newalert নামে একটি এলার্ট তৈরি করেছি একটি এঁরর দেখানোর জন্যঃ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3 id="test"></h3>

<script>

try {

newalert("আপনাকে স্বাগতম!");

}

catch(error) {

document.getElementById("test").innerHTML = error.message;

}

</script>

</body>

</html>

Copy

### ফলাফল

### newalert is not defined

catch ব্লক **newalert** কে error হিসেবে ধরবে এবং হ্যান্ডেল করতে কোড এক্সিকিউট করবে।

## জাভাস্ক্রিপ্টে try এবং catch

এক্সিকিউশনের সময় **try** স্টেটমেন্ট একটি কোডের ব্লককে এঁররের জন্য টেস্ট করতে সাহায্য করে।

যদি try ব্লকে কোন এঁরর ঘটে তাহলে **catch** স্টেটমেন্ট একটি কোডের ব্লককে এক্সিকিউট করতে সাহায্য করে।

জাভাস্ক্রিপ্ট **try** এবং **catch** স্টেটমেন্টের গঠনপ্রণালীঃ

try {

Block of code to try

}

catch(err) {

Block of code to handle errors

}

Copy

## throw স্টেটমেন্ট

**throw** স্টেটমেন্ট একটি কাস্টম এঁরর তৈরি করতে সাহায্য করে।

এর মানে হচ্ছে আপনি একটি এক্সসেপ্সন তৈরি করতে পারেন।

আপনি try এবং catch এর সাথে throw একত্রে ব্যবহার করলে আপনি একটি

ইহা একটি স্ট্রিং, নম্বর, বুলিয়ান অথবা একটি অবজেক্ট হতে পারেঃ

throw "সঠিক নয়"; // টেক্সট throw করবে

throw 231; // নম্বর throw করবে

Copy

## ইনপুট ভ্যালিডেশন

এই উদাহরণে try এবং catch এর সাথে throw এর ব্যবহার দেখানো হয়েছেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>দয়া করে 12 থেকে 17 এর মধ্যে যে কোনো একটি সংখ্যা ইনপুট দিন :</h3>

<input id="test" type="text">

<button type="button" onclick="myFunc()"> ইনপুট যাচাই </button>

<p id="msg"></p>

<script>

function myFunc() {

var message, a;

message = document.getElementById("msg");

message.innerHTML = "";

a = document.getElementById("test").value;

try {

if(a == "") throw "খালি";

if(isNaN(a)) throw " নাম্বার না";

a = Number(a);

if(a < 12) throw "অনেক কম";

if(a > 17) throw "অনেক বেশি";

}

catch(error) {

message.innerHTML = "ইনপুট " + error;

}

}

</script>

</body>

</html

### ফলাফল

### দয়া করে 12 থেকে 17 এর মধ্যে যে কোনো একটি সংখ্যা ইনপুট দিন :

ইনপুট খালি

## এইচটিএমএল ভ্যালিডেশন

আধুনিক ব্রাউজারে জাভাস্ক্রিপ্ট এবং এইচটিএমএল এট্রিবিউট ব্যবহার করে ভ্যালিডেশন করা হয়ঃ

<input id="test" type="number" min="3" max="9" step="1">

Copy

এই টিউটোরিয়ালের পরবর্তী পরিচ্ছেদে [ফর্ম ভ্যালিডেশন](http://www.sattacademy.org/js/js_validation.php) সম্পর্কে আরো জানবেন।

## finally স্টেটমেন্ট

**finally** স্টেটমেন্ট আপনাকে try এবং catch-এর ফলাফলের উপর নির্ভর করে কোড এক্সিকিউট করতে সাহায্য করেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> দয়া করে 12 থেকে 17 পর্যন্ত যে কোনো একটি সংখ্যা ইনপুট দিন :</h3>

<input id="test" type="text">

<button type="button" onclick="myFunc()"> ইনপুট যাচাই </button>

<p id="msg"></p>

<script>

function myFunc() {

var message, a;

message = document.getElementById("msg");

message.innerHTML = "";

a = document.getElementById("test").value;

try {

if(a == "") throw "খালি";

if(isNaN(a)) throw " নাম্বার না";

a = Number(a);

if(a > 12) throw "হলো অনেক বেশি";

if(a < 17) throw "হলো অনেক কম";

}

catch(error) {

message.innerHTML = "ইনপুট " + error;

}

finally {

document.getElementById("test").value = "";

}

}

</script>

</body>

</html>

Copy

### ফলাফল

### দয়া করে ৫ থেকে ১০ পর্যন্ত যে কোনো একটি সংখ্যা ইনপুট দিন :

ইনপুট খালি

## কোড ডিবাগিং

প্রোগ্রামিং কোডের এঁরর খোঁজাকে কোড ডিবাগিং বলে।

যখন কোডের মধ্যে এঁরর হবে,আপনি কোন এঁরর মেসেজ পাবেন না। আপনি বুঝতে পারবেন না কোথায় ভুলের জন্য অনুসন্ধান করবেন।

## জাভাস্ক্রিপ্ট ডিবাগার

ডিবাগ করা সহজ নয়। কিন্তু বর্তমানে সকল আধুনিক ব্রাউজারে একটি বিল্ট-ইন ডিবাগার আছে।

ডিবাগারে ব্রেকপয়েন্ট সেট করে আপনি এক্সিকিউশনের সময় ভ্যারিয়েবলের ভ্যালু দেখতে পারেন।

আপনি F12 কী চেপে আপনার ব্রাউজারে ডিবাগিং মেন্যু থেকে কনসোল চালু করতে পারেন।

## console.log()

console.log() মেথড ব্যবহার করে আপনি আপনার কোডের বিভিন্ন ভ্যালু দেখতে পারবেনঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h2>আমার প্রথন ওয়েব পেজ</h2>

<p>F12 এর সাহায্যে আপনার ব্রাউজারে (Chrome, IE, Firefox) ডিবাগিং সক্রিয় করুন , ডিবাগিং মেন্যুর "Console" অপশনটি সিলেক্ট করুন</p>

<script>

a = 3;

b = 7;

c = a + b;

console.log(c);

</body>

</html>

Copy

### ফলাফল

## আমার প্রথন ওয়েব পেজ

F12 এর সাহায্যে আপনার ব্রাউজারে (Chrome, IE, Firefox) ডিবাগিং সক্রিয় করুন , ডিবাগিং মেন্যুর "Console" অপশনটি সিলেক্ট করুন

## ব্রেকপয়েন্ট সেট করা

ডিবাগার উইন্ডোতে জাভাস্ক্রিপ্টের কোডের মধ্যে আপনি ব্রেকপয়েন্ট সেট করতে পারবেন। ব্রেকপয়েন্টে জাভাস্ক্রিপ্ট এক্সিকিউশন বন্ধ করে আপনাকে জাভাস্ক্রিপ্টের ভ্যালু পরীক্ষা করতে সাহায্য করবে।

ভ্যালু পরীক্ষা করার পর আপনি আবার প্লে বাটনে ক্লিক করে কোডের এক্সিকিউশন চালু করতে পারবেন।

# জাভাস্ক্রিপ্ট জেসন(JSON)

[« পূর্ববর্তী](https://www.sattacademy.com/js/js_errors.php)

[পরবর্তী »](https://www.sattacademy.com/js/js_hoisting.php)

ডাটার সংরক্ষন এবং ডাটার আদান-প্রদানের একটি পদ্ধতি হলো জেসন।

সার্ভার থেকে ওয়েবপেজে ডাটা পাঠানোর জন্যও জেসন ব্যবহার করা হয়।

## জেসন কি?

* জেসন হচ্ছে জাভাস্ক্রিপ্ট অবজেক্ট নোটেশন(JSON - **J**ava**S**cript **O**bject **N**otation)
* জেসন হচ্ছে ডাটা আদান-প্রদানের একটি সুবিধাজনক পদ্ধতি
* জেসন সকল ল্যাংগুয়েজে ব্যবহারযোগ্য**\***
* জেসন স্ব-বর্ণনামূলক এবং এটি বুঝতেও সহজ

\*জেসনের গঠনপ্রণালী জাভাস্ক্রিপ্ট অবজেক্ট নোটেশন হতে পাওয়া। যেকোন প্রোগ্রামিং ল্যাংগুয়েজে জেসন ফরমেটে কোড লিখা যায়।

## গঠনপ্রণালী

জেসনের গঠনপ্রণালী ব্যবহার করে একটি developer অবজেক্ট তৈরি করা হয়েছেঃ ৩জন ডেভেলপারের বিবরণী(অবজেক্ট) একটি অ্যারেতে রাখা হয়েছেঃ

### উদাহরণ

{

"members":[

{"firstName":"Motiur", "lastName":"Rahman"},

{"firstName":"saleh", "lastName":"ahmed"},

{"firstName":"tamjid", "lastName":"hasan"}

]

}

Copy

জাভাস্ক্রিপ্ট অবজেক্ট তৈরির গঠনপ্রণালী এবং জেসনের গঠনপ্রণালী একই।

এই কারণে জেসন ডাটাকে একটি জাভাস্ক্রিপ্ট প্রোগ্রাম খুব সহজেই একটি জাভাস্ক্রিপ্ট অবজেক্টে রুপান্তর করতে পারে।

## JSON গঠনের নিয়ম

* ডাটা name/value আকারে থাকে
* ডাটাকে কমার(,) মাধ্যমে পৃথক করা হয়
* দ্বিতীয় বন্ধনীর ভিতরে অবজেক্ট রাখা হয়
* তৃতীয় বন্ধনীর ভিতরে অ্যারে রাখা হয়

## জেসন ডাটা

জাভাস্ক্রিপ্ট অবজেক্ট প্রোপার্টির মত জেসন ডাটা name/value এর জোড়া আকারে লিখা হয়।

"firstName":"tamim"

Copy

জেসনের নাম লিখতে ডাবল কোটেশন("") দেওয়া বাধ্যতামূলক।

## জেসন অবজেক্ট

জেসন অবজেক্ট দ্বিতীয় বন্ধনীর ভিতরে লেখা হয়। জাভাস্ক্রিপ্ট অবজেক্টের মত জেসন অবজেক্টেও একাধিক name/value এর জোড়া থাকতে পারেঃ

{"firstName":"Tamjid", "lastName":"hasan"}

Copy

## জেসন অ্যারে

জেসন অ্যারে তৃতীয় বন্ধনীর মাধ্যমে লিখা হয়। জাভস্ক্রিপ্টের মত জেসন অ্যারেও অবজেক্ট ধারণ করতে পারেঃ

"members":[

{"firstName":"Tamjid", "lastName":"hasan"},

{"firstName":"Motiur", "lastName":"Rahman"},

{"firstName":"saleh", "lastName":"ahmed"}

]

Copy

## জেসন টেক্সটকে জাভাস্ক্রিপ্ট অবজেক্টে রূপান্তর

ওয়েব সার্ভার থেকে ডাটা ওয়েব পেজে দেখানোর জন্য জেসন ব্যবহার করা হয়।

প্রথমে জেসন ডাটাকে জাভাস্ক্রিপ্ট স্ট্রিংয়ের মধ্যে রাখতে হবে,তারপর ঐ স্ট্রিংকে জাভাস্ক্রিপ্ট অবজেক্টে রুপান্তর করতে জাভাস্ক্রিপ্টের JSON.parse() ফাংশনটি ব্যবহার করুনঃ

### উদাহরণ

var information = '{ "members" : [' +

'{"firstName":"Motiur", "lastName":"Rahman"},' +

'{"firstName":"Tamjid", "lastName":"hasan"},' +

'{"firstName":"saleh", "lastName":"ahmed"} ]}';

var obj = JSON.parse(information);

Copy

তারপর নতুন জাভাস্ক্রিপ্ট অবজেক্ট ব্যবহার করুনঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h2> JSON স্ট্রিং থেকে অবজেক্ট তৈরী করা</h2>

<p id="test"></p>

<script>

var information = '{"members":[' +

'{"firstName":"আজিজুর","lastName":"রহমান" },' +

'{"firstName":" তামিম","lastName":" তাহমিদ " },' +

'{"firstName":"শাপলা","lastName":"আজিজ" }]}';

obj = JSON.parse(information);

document.getElementById("test").innerHTML =

obj.members[1].firstName + " " + obj.members[1].lastName;

</script>

</body>

</html>

Copy

### ফলাফল

## JSON স্ট্রিং থেকে অবজেক্ট তৈরী করা

তামিম তাহমিদ

# জাভাস্ক্রিপ্ট হয়েস্টিং

জাভাস্ক্রিপ্টের একটি ডিফল্ট আচরণ হচ্ছে হয়েস্টিং।

জাভাস্ক্রিপ্টে, একটি ভ্যারিয়েবলকে ডিক্লেয়ার করার পূর্বে ব্যবহার করা যায়।

নিচের দুইটি উদাহরণ একই ফলাফল দেখাবেঃ

### উদাহরণ-1

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

a = 20; // Assign 20 to a

element = document.getElementById("test"); // Find an element

element.innerHTML = a; // Display a in the element

var a; // Declare a

</script>

</body>

</html>

Copy

### ফলাফল

20

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var a; // a ঘোষনা করা হয়েছ

a = 20; // a এর মধ্যে 20 রাখা হয়েছে

element = document.getElementById("test"); // এটা একটা এলিমেন্ট

element.innerHTML = a; // a এর এলিমেন্ট প্রদর্শণ করা হলো

</script>

</body>

</html>

Copy

### ফলাফল

20

এটা বুঝার জন্য আপনাকে হয়েস্টিং("hoisting") বিষয়টি বুঝতে হবে।

জাভাস্ক্রিপ্ট ডিফল্টভাবে সকল ডিক্লেয়ারেশনকে তার নিজ নিজ স্কোপের উপরে নিয়ে যায়,একেই হয়েস্টিং বলা হয়।

## ইনিশিয়ালাইজেশন হয়েস্টিং-এর অন্তর্ভুক্ত নয়

জাভাস্ক্রিপ্টে হয়েস্টিং শুধুমাত্র ডিক্লেয়ারেশনের জন্য প্রযোজ্য। ইনিশিয়ালাইজেশন হয়েস্টিং-এর অন্তর্ভুক্ত নয়।

নিচের দুইটি উদাহরণ দেখিঃ

### উদাহরণ-1

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var a = 3; // a এর মান নির্ধারন করা হয়েছে

var b = 8; // b এর মান নির্ধারন করা হয়েছে

element = document.getElementById("test"); // এটা একটা এলিমেন্ট

element.innerHTML = a + " " + b; // a এবং b প্রদর্শন করা হয়েছে

</script>

</body>

</html>

Copy

### ফলাফল

38

### উদাহরণ-2

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var a = 3; // a এর মান নির্ধারন করা হয়েছে

element = document.getElementById("test"); // এটা একটা এলিমেন্ট

element.innerHTML = "a হল " + a + " এবং b হল " + b; // a এবং b প্রদর্শন করা হয়েছে

var b = 8; // b এর মান নির্ধারন করা হয়েছে

</script>

</body>

</html>

Copy

### ফলাফল

a হল 3 এবং b হল undefined

দ্বিতীয় উদাহরণে b এর ভ্যালু undefined দেখিয়েছে। কারণ জাভাস্ক্রিপ্ট শুধুমাত্র ডিক্লেয়ারেশনকে উপরে নিয়েছে, এসাইনকৃত ভ্যালুকে উপরে নেয়নি। তাই b এর ভ্যালু undefined দেখানো হয়েছে।

দ্বিতীয় উদাহরণটি নিচের উদাহরণের মতইঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var a = 6; // Initialize a

var b; // Declare b

element = document.getElementById("test"); // Find an element

element.innerHTML = a + " " + b; // Display a and b

b = 9; // Assign 9 to b

</script>

</body>

</html>

Copy

### ফলাফল

6 undefined

সবসময় ভ্যারিয়েবলকে স্কোপের উপরে ডিক্লেয়ার করা একটি ভাল অভ্যাস।

### "use strict";

জাভাস্ক্রিপ্ট কোডে "use strict"; ব্যবহার একটি ভালো দিক। ইহা প্রোগ্রামারদের সিন্টেক্সগত ভুল, অনিচ্ছাকৃত গ্লোবাল ভ্যারিয়েবল তৈরি এবং অন্যান্য সমস্যা থেকে রক্ষা করে। এই বিষয়গুলো জাভাস্ক্রিপ্টে কোন এঁরর দেখায় না, কিন্তু "use strict"; ব্যবহার করলে ইহা এঁরর হিসেবে গণ্য হয়।

একটি স্ক্রিপ্ট এবং প্রতিটি ফাংশনের শুরুতে "use strict"; ডিক্লেয়ার করা যায়।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h1>"use strict"</h1>

<h3>ভেরিয়েবল ডিক্লার করা ব্যতিত ভেরিয়েবল ব্যবহার করতে অনুমতি দেয় না।</h3>

<p>আপনার ব্রাউজারে এরর রিপোর্ট দেখার জন্য ডিবাগিং (F12) এক্টিভ করুন।</p>

<script>

"use strict";

a = "tamim"; // এটি একটি এরর দেখাবে (a ডিফাইন করা হয়নি).

</script>

</body>

</html>

Copy

### ফলাফল

# "use strict"

### ভেরিয়েবল ডিক্লার করা ব্যতিত ভেরিয়েবল ব্যবহার করতে অনুমতি দেয় না।

আপনার ব্রাউজারে এরর রিপোর্ট দেখার জন্য ডিবাগিং (F12) এক্টিভ করুন।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h2>গ্লোবাল "use strict" ডিক্লারেশন</h2>

<p>আপনার ব্রাউজারে এরর রিপোর্ট দেখার জন্য ডিবাগিং (F12) এক্টিভ করুন।</p>

<script>

"use strict";

myFunc();

function myFunc() {

a = "tamim"; // এটি একটি এরর দেখাবে (a ডিফাইন করা হয়নি)

}

</script>

</body>

</html>

ফলাফল

## গ্লোবাল "use strict" ডিক্লারেশন

আপনার ব্রাউজারে এরর রিপোর্ট দেখার জন্য ডিবাগিং (F12) এক্টিভ করুন।

"use strict"; যদি স্ক্রিপ্টের শুরুতে ব্যবহার করা হয় তবে ইহা স্ক্রিপ্টের সকল কোডের জন্য প্রযোজ্য হবে এবং ফাংশনের ভিতরে ব্যবহার করা হলে ইহা শুধুমাত্র ফাংশনের কোডের জন্য প্রযোজ্য।

# জাভাস্ক্রিপ্ট বুলিয়ান

জাভাস্ক্রিপ্ট বুলিয়ান নিচের দুইটি ভ্যালুর যেকোন একটি প্রদর্শন করেঃ **true** অথবা **false** ।

## বুলিয়ান ভ্যালু

প্রোগ্রামিং-এ আপনার এমন একটি ডাটা টাইপ দরকার হবে যেখানে শুধুমাত্র নিচের যেকোন একটি ভ্যালুর প্রয়োজন হবে, যেমনঃ

* YES / NO
* ON / OFF
* TRUE / FALSE

এই কারনে, জাভাস্ক্রিপ্টের একটি **বুলিয়ান** ডাটা টাইপ আছে। এটি শুধুমাত্র **true** অথবা **false** ভ্যালু ধারণ করে।

## Boolean() ফাংশন

একটি এক্সপ্রেশন সত্য কিনা যাচাই করার জন্য আপনি Boolean() ব্যবহার করতে পারেনঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> বুলিয়ান(4 > 2) এর ভ্যালু প্রদর্শিত হবেঃ</h3>

<button onclick="myFunc()">চেষ্টা করি</button>

<p id="test"></p>

<script>

function myFunc() {

document.getElementById("test").innerHTML = Boolean(4 > 2);

}

</script>

</body>

</html>

Copy

### ফলাফল

### বুলিয়ান(4 > 2) এর ভ্যালু প্রদর্শিত হবেঃ

true

আরো সহজভাবেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> 4 > 2 এর ভ্যালুকে প্রদর্শন করে </h3>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

function myFunc() {

document.getElementById("test").innerHTML = 4 > 2;

}

</script>

</body>

</html>

Copy

### ফলাফল

### 4 > 2 এর ভ্যালুকে প্রদর্শন করে

true

## কম্প্যারিজন এবং কন্ডিশন অপারেটর

জাভাস্ক্রিপ্ট কম্প্যারিজন অধ্যায়ে কম্প্যারিজন অপারেটর সম্পর্কে সম্পূর্ণ আলোচনা করা হয়েছে।

জাভাস্ক্রিপ্ট কন্ডিশন বা শর্ত পরিচ্ছেদে কন্ডিশনাল স্ট্যাটমেন্ট সম্পর্কে সম্পূর্ণ আলোচনা করা হয়েছে।

উদাহরণঃ

|  |  |  |
| --- | --- | --- |
| **অপারেটর** | **বর্ণনা** | **উদাহরণ** |
| == | equal to(সমান) | if (day == "Monday") |
| > | greater than(বড়) | if (salary > 9000) |
| < | less than(ছোট) | if (age < 18) |

## সকল সত্যিকারের ভ্যালু হবে ****True****

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var b1 = Boolean(70);

var b2 = Boolean(1.99);

var b3 = Boolean(-40);

var b4 = Boolean(3 + 5 + 2.6);

var b5 = Boolean("Welcome");

var b6 = Boolean('false');

document.getElementById("test").innerHTML =

"70 হলো " + b1 + "" +

"1.99 হলো " + b2 + "" +

"-40 হলো " + b3 + "" +

"0 ছাড়া সকল এক্সপ্রেশন হলো " + b4 + "" +

"খালি ছাড়া সকল স্ট্রিং হলো " + b5 + "" +

"এমনকি স্ট্রিং 'false' হলো " + b6;

</script>

</body>

</html>

Copy

### ফলাফল

70 হলো true  
1.99 হলো true  
-40 হলো true  
0 ছাড়া সকল এক্সপ্রেশন হলো true  
খালি ছাড়া সকল স্ট্রিং হলো true  
এমনকি স্ট্রিং 'false' হলো true

## সত্যিকার নম্বর ছাড়া সকল কিছুর ভ্যালু হবে False

### উদাহরণ

বুলিয়ান ভ্যালু **0** হচ্ছে **false**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> ০ এর বুলিয়ান ভ্যালু প্রদর্শন করেঃ </h3>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

function myFunc() {

var a = 0;

document.getElementById("test").innerHTML = Boolean(a); // false রিটার্ন করবে

}

</script>

</body>

</html>

Copy

### ফলাফল

### ০ এর বুলিয়ান ভ্যালু প্রদর্শন করেঃ

false

### উদাহরণ

বুলিয়ান ভ্যালু **-0** হচ্ছে false ।

বুলিয়ান ভ্যালু **0** এর নিচে সব false

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> -0 এর বুলিয়ান ভ্যালু প্রদর্শন করেঃ </h3>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

function myFunc() {

var a = -0;

document.getElementById("test").innerHTML = Boolean(a); // false রিটার্ন করবে

}

</script>

</body>

</html>

Copy

### ফলাফল

### -0 এর বুলিয়ান ভ্যালু প্রদর্শন করেঃ

false

### উদাহরণ

"" (খালি স্ট্রিং) এর বুলিয়ান ভ্যালু হচ্ছে false

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> "" এর বুলিয়ান ভ্যালু প্রদর্শন করেঃ</h3>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

function myFunc() {

var a = "";

document.getElementById("test").innerHTML = Boolean(a); // false রিটার্ন করবে

}

</script>

</body>

</html>

Copy

### ফলাফল

### "" এর বুলিয়ান ভ্যালু প্রদর্শন করেঃ

false

### উদাহরণ

undefined এর বুলিয়ান ভ্যালু হচ্ছে false

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> undefined এর বুলিয়ান ভ্যালু প্রদর্শন করেঃ</h3>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

function myFunc() {

var a;

document.getElementById("test").innerHTML = Boolean(a); // false রিটার্ন করবে

}

</script>

</body>

</html>

Copy

### ফলাফল

### undefined এর বুলিয়ান ভ্যালু প্রদর্শন করেঃ

false

### উদাহরণ

null এর বুলিয়ান ভ্যালু হচ্ছে false

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> null এর বুলিয়ান ভ্যালু প্রদর্শন করেঃ</h3>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

function myFunc() {

var a = null;

document.getElementById("test").innerHTML = Boolean(a); // false রিটার্ন করবে

}

</script>

</body>

</html>

Copy

### ফলাফল

### null এর বুলিয়ান ভ্যালু প্রদর্শন করেঃ

false

### উদাহরণ

false এর বুলিয়ান ভ্যালু হচ্ছে false

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> false এর বুলিয়ান ভ্যালু প্রদর্শন করেঃ</h3>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

function myFunc() {

var a = false;

document.getElementById("test").innerHTML = Boolean(a); // false রিটার্ন করবে

}

</script>

</body>

</html>

Copy

### ফলাফল

### false এর বুলিয়ান ভ্যালু প্রদর্শন করেঃ

false

### উদাহরণ

NaN এর বুলিয়ান ভ্যালু হচ্ছে false

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> NaN এর বুলিয়ান ভ্যালু প্রদর্শন করেঃ</h3>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

function myFunc() {

var a = 5/"M";

document.getElementById("test").innerHTML = Boolean(a); // false রিটার্ন করবে

}

</script>

</body>

</html>

Copy

### ফলাফল

### NaN এর বুলিয়ান ভ্যালু প্রদর্শন করেঃ

false

## বুলিয়ান প্রোপার্টি এবং মেথড

true এবং false এর মত মৌলিক ভ্যালুর প্রোপার্টি এবং মেথড থাকে না(কারন তারা অবজেক্ট না)।

কিন্তু জাভাস্ক্রিপ্টে, মৌলিক ভ্যালুতে মেথড এবং প্রোপার্টি থাকে কারন যখন মেথড এবং প্রোপার্টি এক্সিকিউটিং হয় তখন জাভাস্ক্রিপ্ট মৌলিক ভ্যালুকে অবজেক্ট হিসেবে বিবেচনা করে।

## সম্পূর্ণ বুলিয়ান রেফারেন্স

সম্পূর্ণ বুলিয়ান রেফারেন্সের জন্য আমাদের [সম্পূর্ণ জাভাস্ক্রিপ্ট বুলিয়ান রেফারেন্স](https://www.sattacademy.com/jsref/jsref_obj_boolean.php) পড়ুন।

## জাভাস্ক্রিপ্ট বুলিয়ান(Booleans)

জাভাস্ক্রিপ্ট বুলিয়ানের দুইটি ভ্যালু থাকেঃ **True** এবং **False**।

## Boolean() ফাংশন

Boolean() ফাংশন ব্যবহার করে আপনি একটি এক্সপ্রেশনের সত্যতা যাচাই করতে পারেনঃ

### উদাহরণ

<!DOCTYPE html>

<html>

<head>

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

document.getElementById("test").innerHTML = Boolean(4 > 2);

</script>

</body>

</html>

Copy

### ফলাফল

true

অথবা আরো সহজেঃ

### উদাহরণ

<!DOCTYPE html>

<html>

<head>

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

document.getElementById("test").innerHTML = (4 > 2);

</script>

</body>

</html>

Copy

### ফলাফল

true

বুলিয়ান সম্পর্কে আরো জানতে আমাদের [জাভাস্ক্রিপ্ট বুলিয়ান টিউটোরিয়াল](https://www.sattacademy.com/js/js_booleans.php) পড়ুন।

## বুলিয়ান প্রোপার্টি

|  |  |
| --- | --- |
| **প্রোপার্টি** | **বর্ণনা** |
| constructor | যে ফাংশন জাভাস্ক্রিপ্ট বুলিয়ান প্রোটোটাইপ তৈরি করে তা রিটার্ন করে। |
| prototype | বুলিয়ান প্রোটোটাইপে প্রোপার্টি এবং মেথড যোগ করে। |

## বুলিয়ান মেথড

|  |  |
| --- | --- |
| **মেথড** | **বর্ণনা** |
| toString() | বুলিয়ান ভ্যালুকে স্ট্রিং এ পরিনত করে এবং রেজাল্ট রিটার্ন করে। |
| valueOf() | একটি বুলিয়ানের প্রিমিটিভ ভ্যালু রিটার্ন করে। |

# জাভাস্ক্রিপ্টের কম্প্যারিজন এবং লজিক্যাল অপারেটর

true অথবা false যাচাই করার জন্য কম্প্যারিজন এবং লজিক্যাল অপারেটর ব্যবহার করা হয়।

## কম্প্যারিজন অপারেটর

কম্প্যারিজন অপারেটর ভ্যারিয়েবল এবং ভ্যালুর মধ্যে সাদৃশ্য অথবা পার্থক্য নির্ধারন করার জন্য ব্যবহার করা হয়।

দেওয়া আছে **x = 5**, নিম্নোক্ত টেবিলে কম্প্যারিজন অপারেটর ব্যাখ্যা করা হয়েছেঃ

|  |  |  |  |
| --- | --- | --- | --- |
| **অপারেটর** | **বর্ণনা** | **তুলনা** | **রিটার্ন করে** |
| == | সমান | x == 8 | false |
| x == 5 | true |
| x == "5" | true |
| === | একই ভ্যালু এবং একই টাইপ | x === 5 | true |
| x === "5" | false |
| != | সমান নয় | x != 8 | true |
| !== | ভ্যালু সমান নয় এবং টাইপ একই নয় | x !== 5 | false |
| x !== "5" | true |
| x !== 8 | true |
| > | বড় | x > 8 | false |
| < | ছোট | x < 8 | true |
| >= | বড় অথবা সমান | x >= 8 | false |
| <= | ছোট অথবা সমান | x <= 8 | true |

## যেভাবে ব্যবহার করবেন

কন্ডিশনাল স্টেটমেন্টে কম্প্যারিজন অপারেটর ব্যবহার করে ভ্যালুর তুলনা করা হয় এবং ফলাফলের উপর নির্ভর করে পরবর্তী কার্যক্রম নির্ধারন করা হয়ঃ

if (age < 18) text = "Too young";

Copy

কন্ডিশনাল স্টেটমেন্ট সম্পর্কে পরবর্তী পরিচ্ছেদে আরো শিখতে পারবেন।

## লজিক্যাল অপারেটর

লজিক্যাল অপারেটর ভ্যারিয়েবল এবং ভ্যালুর মধ্যে লজিক নির্ধারন করে।

দেওয়া আছে **x = 6** এবং **y = 3**, নিম্নোক্ত টেবিলে লজিক্যাল অপারেটর সম্পর্কে ব্যাখ্যা করা হয়েছেঃ

|  |  |  |
| --- | --- | --- |
| **অপারেটর** | **বর্ণনা** | **উদাহরণ** |
| && | এবং | (x < 10 && y > 1) true রিটার্ন করে |
| || | অথবা | (x == 5 || y == 5) false রিটার্ন করে |
| ! | না | !(x == y) true রিটার্ন করে |

## কন্ডিশনাল(Ternary) অপারেটর

জাভাস্ক্রিপ্টে একটি কন্ডিশনাল অপারেটর রয়েছে যা শর্তের উপর ভিত্তি করে ভ্যারিয়েবলে ভ্যালু এসাইন করে।

### গঠনপ্রণালী

variablename = (condition) ? value1:value2

Copy

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> আপনার বয়স সাবমিট করুন এবং বাটনে ক্লিক করুন :</h3>

<input id="age" value="18" />

<button onclick="myFunc()">সাবমিট </button>

<p id="test"></p>

<script>

function myFunc() {

var age, voteable;

age = document.getElementById("age").value;

voteable = (age < 18) ? " অপ্রাপ্ত বয়স্ক ":" প্রাপ্ত বয়স্ক";

document.getElementById("test").innerHTML = voteable + " ভোট দেওয়ার জন্য";

}

</script>

</body>

</html>

Copy

### ফলাফল

### আপনার বয়স সাবমিট করুন এবং বাটনে ক্লিক করুন :

ভোট দেওয়ার জন্য প্রাপ্ত বয়স্ক।

যদি ভ্যারিয়েবল age এর ভ্যালু 18 এর নিচে হয় তাহলে ভ্যারিয়েবলের ভ্যালু হবে "Can not drive" অন্যথায় ভ্যারিয়েবলের ভ্যালু হবে "Can drive" ।

## বিভিন্ন টাইপের মধ্যে তুলনা

বিভিন্ন টাইপের ডাটার মধ্যে তুলনা করলে অপ্রত্যাশিত ফলাফল দেখাতে পারে।

স্ট্রিংকে একটি নম্বরের সাথে তুলনা করলে জাভাস্ক্রিপ্ট একটি স্ট্রিংকে নম্বরে রুপান্তর করে। এম্পটি স্ট্রিং 0 তে রুপান্তর হয়। একটি টেক্সট স্ট্রিং "NaN" এ রুপান্তর হয় যা সব সময় false হয়।

|  |  |
| --- | --- |
| **উদাহরণ** | **ভ্যালু** |
| 2 < 12 | true |
| 2 < "12" | true |
| 2 < "Azizur" | false |
| 2 > "Rahman" | false |
| 2 == "Azizur" | false |
| "2" < "12" | false |
| "2" > "12" | true |
| "2" == "12" | false |

যখন দুইটি স্ট্রিংকে তুলনা করা হয়, তখন "2" চেয়ে "12" ছোট হবে। কারন বর্ণানুক্রমে 2 এর চেয়ে 1 ছোট।

একটি সঠিক ফলাফলের জন্য, ভ্যারিয়েবলকে তুলনা করার আগে একটি সঠিক টাইপে রুপান্তর করতে হবেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> আপনার বয়স সাবমিট করুন এবং বাটনে ক্লিক করুন :</h3>

<input id="age" value="18" />

<button onclick="myFunc()">সাবমিট </button>

<p id="test"></p>

<script>

function myFunc() {

var age, voteable;

age = Number(document.getElementById("age").value);

if(isNaN(age)){

voteable = "ভুল ইনপুট";

}else{

voteable = (age < 18) ? " অপ্রাপ্ত বয়স্ক ":" প্রাপ্ত বয়স্ক";

}

document.getElementById("test").innerHTML = voteable;

}

</script>

</body>

</html>

Copy

### ফলাফল

### আপনার বয়স সাবমিট করুন এবং বাটনে ক্লিক করুন :

প্রাপ্ত বয়স্ক

## জাভাস্ক্রিপ্ট বিটওয়াইজ অপারেটর

বিট অপারেটর ৩2-বিট নম্বরে কাজ করে।

যেকোনো নিউমেরিক অপারেন্ডকে ৩2-বিট নম্বরে রুপান্তর হয়।

এই ফলাফল জাভাস্ক্রিপ্ট নম্বরে রুপান্তর হয়।

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **অপারটর** | **বর্ণনা** | **উদাহরণ** | **এর মত** | **ফলাফল** | **ডেসিমাল** |
| & | AND | x = 5 & 1 | 0101 & 0001 | 0001 | 1 |
| | | OR | x = 5 | 1 | 0101 | 0001 | 0101 | 5 |
| ~ | NOT | x = ~ 5 | ~0101 | 1010 | 10 |
| ^ | XOR | x = 5 ^ 1 | 0101 ^ 0001 | 0100 | 4 |
| << | Left shift | x = 5 << 1 | 0101 << 1 | 1010 | 10 |
| >> | Right shift | x = 5 >> 1 | 0101 >> 1 | 0010 | 2 |

উপরের টেবিল ৪ বিট unsigned উদাহরণ ব্যবহার করেছে। কিন্তু জাভাস্ক্রিপ্ট ৩2-বিট নম্বর ব্যবহার করে।   
এই কারনে জাভাস্ক্রিপ্ট, ~ ৫ কে 1০ রিটার্ন করে না এটি -৬ রিটার্ন করেঃ   
~00000000000000000000000000000101 এর রিটার্ন হবে 11111111111111111111111111111010

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>বিটওয়াইজ অপারেটর " & " দ্বারা AND বিটওয়াইজকে বুঝায় ।</h3>

<button onclick="myFunc()">সাবমিট </button>

<p id="test"></p>

<script>

function myFunc() {

document.getElementById("test").innerHTML = 5 & 1;

}

</script>

</body>

</html>

Copy

### ফলাফল

### বিটওয়াইজ অপারেটর " & " দ্বারা AND বিটওয়াইজকে বুঝায় ।

1

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> বিটওয়াইজ অপারেটর " | " দ্বারা or বিটওয়াইজকে বুঝায়</h3>

<button onclick="myFunc()"> সাবমিট</button>

<p id="test"></p>

<script>

function myFunc() {

document.getElementById("test").innerHTML = 5 | 1;

}

</script>

</body>

</html>

Copy

### ফলাফল

### বিটওয়াইজ অপারেটর " | " দ্বারা or বিটওয়াইজকে বুঝায়

5

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> বিটওয়াইজ অপারেটর " >> " দ্বারা right shift বিটওয়াইজকে বুঝায় </h3>

<button onclick="myFunc()"> সাবমিট </button>

<p id="test"></p>

<script>

function myFunc() {

document.getElementById("test").innerHTML = 5 >> 1;

}

</script>

</body>

</html>

Copy

### ফলাফল

### বিটওয়াইজ অপারেটর " >> " দ্বারা right shift বিটওয়াইজকে বুঝায়

2

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> left shift অপারেটর দ্বারা left shift বিটওয়াইজ কে বুঝায় ।</h3>

<button onclick="myFunc()"> চেষ্টা করি </button>

<p id="test"></p>

<script>

function myFunc() {

document.getElementById("test").innerHTML = 5 << 1;

}

</script>

</body>

</html>

Copy

### ফলাফল

### left shift অপারেটর দ্বারা left shift বিটওয়াইজ কে বুঝায় ।

10

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> বিটওয়াইজ অপারেটর " ~ " দ্বারা বিটওয়াইজ NOT কে বুঝায় ।</h3>

<button onclick="myFunc()">সাবমিট</button>

<p id="test"></p>

<script>

function myFunc() {

document.getElementById("test").innerHTML = ~ 5;

}

</script>

</body>

</html>

Copy

### ফলাফল

### বিটওয়াইজ অপারেটর " ~ " দ্বারা বিটওয়াইজ NOT কে বুঝায় ।

-6

# জাভাস্ক্রিপ্ট If...Else স্টেটমেন্ট

কন্ডিশনাল স্টেটমেন্ট বিভিন্ন শর্তের উপর ভিত্তি করে বিভিন্ন কার্য সম্পাদন করে।

## কন্ডিশনাল স্টেটমেন্ট

কোড লিখার সময় আপনি মাঝে মাঝে বিভিন্ন সিদ্ধান্তের উপর বিভিন্ন কার্য সম্পাদন করতে চাইবেন।

এটা করার জন্য আপনি আপনার কোডে কন্ডিশনাল স্টেটমেন্ট ব্যবহার করতে পারেন।

জাভাস্ক্রিপ্টে নিম্নোক্ত কন্ডিশনাল স্টেটমেন্ট আছেঃ

* **if**-এর শর্ত true হলে ব্লকের কোড সম্পাদন হবে।
* যদি একই শর্ত false হয় তখন **else** এর মধ্যে থাকা কোড এক্সিকিউট হবে।
* যদি প্রথম শর্ত false হয় তাহলে নতুন শর্ত টেস্ট করার জন্য **else if** ব্যবহার করুন।
* যখন এক্সিকিউশনের জন্য বিকল্প অনেক কোড নির্দেশ করতে তখন **switch** ব্যবহার করুন।

## if স্টেটমেন্ট

**if**-এর শর্ত যদি true হয় তাহলে ব্লকের মধ্যে জাভাস্ক্রিপ্ট কোড এক্সিকিউট হবে।

### গঠনপ্রণালী

if (condition) {

যদি শর্ত true হয় তাহলে এই কোড এক্সিকিউট হবে

}

Copy

**if** লেখাটি ছোট হাতের লিখতে হবে। বড় হাতের(If অথবা IF) লিখলে জাভাস্ক্রিপ্ট ভুল দেখাবে।

### উদাহরণ

যদি সময় 18:00(সন্ধ্যা ০৬ টা) এর নিচে হয় তাহলে "Good day" শুভেচ্ছা জানাবেঃ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>যদি সময় ১৮.০০ এর নিচে হয় তাহলে "শুভ দিন " শুভেচ্ছা জানাবেঃ</h3>

<p id="test"></p>

<script>

if (new Date().getHours() < 18) {

document.getElementById("test").innerHTML = "শুভ দিন !";

}

</script>

</body>

</html>

Copy

### ফলাফল

### যদি সময় ১৮.০০ এর নিচে হয় তাহলে "শুভ দিন " শুভেচ্ছা জানাবেঃ

শুভ দিন !

## else স্টেটমেন্ট

যদি শর্ত false হয় তাহলে **else** এর মধ্যে থাকা কোড এক্সিকিউট হয়।

if (condition) {

যদি শর্ত true হয় তাহলে এই কোড এক্সিকিউট হবে

}

else {

যদি শর্ত false হয় তাহলে এই কোড এক্সিকিউট হবে

}

Copy

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> নিচের বাটনে ক্লিক করলে সময়-ভিত্তিক শুভেচ্ছা প্রদর্শন করবেঃ </h3>

<button onclick="myFunc()"> চেষ্টা করি </button>

<p id="test"></p>

<script>

function myFunc() {

var hour = new Date().getHours();

var text;

if (hour < 18) {

text = "শুভ দিন";

} else {

text = "শুভ সন্ধ্যা";

}

document.getElementById("test").innerHTML = text;

}

</script>

</body>

</html>

Copy

### ফলাফল

### নিচের বাটনে ক্লিক করলে সময়-ভিত্তিক শুভেচ্ছা প্রদর্শন করবেঃ

শুভ দিন

## else if  স্টেটমেন্ট

যদি প্রথম শর্ত false হয় তাহলে শর্ত যোগ করার জন্য **else if** স্টেটমেন্ট ব্যবহার হয়।

### গঠনপ্রণালী

if (condition1) {

যদি শর্ত-১ true হয় তাহলে এই কোড এক্সিকিউট হবে

}

else if (condition2) {

যদি শর্ত-১ false এবং শর্ত-২ true হয় তাহলে এই কোড এক্সিকিউট হবে

} else {

যদি শর্ত-১ এবং শর্ত-২ false হয় তাহলে এই কোড এক্সিকিউট হবে

}

Copy

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> নিচের বাটনে ক্লিক করলে সময়-ভিত্তিক শুভেচ্ছা পাওয়া যাবেঃ </h3>

<button onclick="myFunc()"> চেষ্টা করি </button>

<p id="test"></p>

<script>

function myFunc() {

var text;

var hour = new Date().getHours();

if (hour < 10) {

text = " শুভ সকাল";

} else if (hour < 20) {

text = " শুভ দিন";

} else {

text = " শুভ সন্ধ্যা";

}

document.getElementById("test").innerHTML = text;

}

</script>

</body>

</html>

Copy

### ফলাফল

### নিচের বাটনে ক্লিক করলে সময়-ভিত্তিক শুভেচ্ছা পাওয়া যাবেঃ

শুভ দিন

# জাভাস্ক্রিপ্ট Switch স্টেটমেন্ট

ভিন্ন ভিন্ন শর্তের উপর ভিত্তি করে ভিন্ন ভিন্ন কার্য সম্পাদনের জন্য Switch স্টেটমেন্ট ব্যবহার করা হয়।

## জাভাস্ক্রিপ্ট Switch স্টেটমেন্ট

অনেকগুলো কোডের ব্লক থেকে এক্সিকিউট করার জন্য একটিকে নির্বাচন করতে Switch স্টেটমেন্ট ব্যবহার করা হয়।

### গঠনপ্রণালী

switch(expression) {

case n:

কোড

break;

case n:

কোড

break;

default:

ডিফল্ট কোড

}

Copy

কাজের প্রক্রিয়াঃ

* এখানে Switch এক্সপ্রেশন একবার মূল্যায়ন করা হয় ।
* এক্সপ্রেশনের ভ্যালুকে প্রত্যেকটি case এর ভ্যালুর সাথে তুলনা করা হয়।
* যদি কোন একটির সাথে মিলে যায় তাহলে ঐ case এর ব্লকটি এক্সিকিউট হবে।

getDay() ব্যবহার করে আমরা সাপ্তাহিক দিনকে নম্বরে(0-6 পর্যন্ত) পেতে পারি।(Sunday=0, Monday=1, Tuesday=2...)

আমরা এই নম্বরকে দিনের নাম পাওয়ার জন্য ব্যবহার করতে পারিঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var day;

switch (new Date().getDay()) {

case 0:

day = "রবিবার";

break;

case 1:

day = " সোমবার";

break;

case 2:

day = "মঙ্গলবার";

break;

case 3:

day = " বুধবার";

break;

case 4:

day = " বৃহস্পতিবার";

break;

case 5:

day = "শুক্রবার";

break;

case 6:

day = "শনিবার";

}

document.getElementById("test").innerHTML = " আজ " + day;

</script>

</body>

</html>

Copy

### ফলাফল

আজ শুক্রবার

## break কিওয়ার্ড

যখন জাভাস্ক্রিপ্ট **break** কি-ওয়ার্ডে পৌছাবে, তখন switch ব্লকটি কাজ করা বন্ধ করে দিবে।

ইহা অন্য কোন Case কে যাচাই করবেনা এবং অতিরিক্ত কোন কোডও এক্সিকিউট করবে না।

যখনই মিল খুজে পাবে তখনি তার কাজ সম্পন্ন হয়ে যাবে এবং এক্সিকিউশন বন্ধ করে দিবে। পরবর্তী কোন case কে যাচাই করবে না।

Break ব্যবহারে এক্সিকিউশনের সময় অনেক কমে যায় কারন যেখানে মিল খুজে পাবে তারপরই এক্সিকিউশন বন্ধ করে দিবে।

Switch ব্লকের সর্বশেষ case-এর break নেওয়ার প্রয়োজন নেই কারণ ব্লক এখানে এমনিতেই বন্ধ হয়ে যাবে।

## ডিফল্ট কিওয়ার্ড

**default** কিওয়ার্ডের মাধ্যমে আমরা কোড রান করাতে পারি যদি কোন case এর সাথে মিল না পাওয়া যায়ঃ

getDay() ব্যবহার করে আমরা সাপ্তাহিক দিনকে নম্বরে(0-6 পর্যন্ত) পেতে পারি।

যদি আজকের দিনটি শনিবার(6) অথবা রবিবার(0) না হয় তবে একটি ডিফল্ট মেসেজ দেখাবেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var text;

switch (new Date().getDay()) {

case 6:

text = " আজ শনিবার";

break;

case 0:

text = " আজ রবিবার";

break;

default:

text = " ছুটির দিনের জন্য অপেক্ষা";

}

document.getElementById("test").innerHTML = text;

</script>

</body>

</html>

Copy

### ফলাফল

ছুটির দিনের জন্য অপেক্ষা

মাঝে মাঝে আপনি একটি switch ব্লকে বিভিন্ন case-এ একই কোড ব্যবহার করতে চাইবেন অথবা একটি সাধারণ ডিফল্টে থেমে যেতে চাইবেন।

নিচের উদাহরণে দেখুন ভিন্ন ভিন্ন case একই code এর ব্লক ব্যবহার করতে পারে এবং ডিফল্ট case টি Switch ব্লকের সর্বশেষ case হওয়ারও প্রয়োজন নেইঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var text;

switch (new Date().getDay()) {

case 1:

case 2:

case 3:

default:

text = " ছুটির দিনের জন্য অপেক্ষা ";

break;

case 4:

case 5:

text = " শীগ্রই ছুটি আসছে";

break;

case 0:

case 6:

text = "এটা ছুটির দিন";

}

document.getElementById("test").innerHTML = text;

</script>

</body>

</html>

Copy

### ফলাফল

শীগ্রই ছুটি আসছে

যদি ডিফল্ট case টি Switch ব্লকের সর্বশেষ case না হয় তবে break দিয়ে শেষ হবে।

# জাভাস্ক্রিপ্ট For লুপ

লুপের মাধ্যমে একগুচ্ছ কোডকে একাধিকবার এক্সিকিউট করানো যায়।

## জাভাস্ক্রিপ্ট লুপ

যদি আপনি একই কোড বহুবার রান করিয়ে প্রত্যেকবার ভিন্ন ভিন্ন ভ্যালু পেতে চান তাহলে লুপ সবচেয়ে সুবিধাজনক পদ্ধতি।

আপনি অ্যারে ব্যবহার করলে অধিকাংশ সময় লুপ ব্যবহার করবেনঃ

### এই কোডগুলো লিখার পরিবর্তেঃ

text += animals[0] + "<br>";

text += animals[1] + "<br>";

text += animals[2] + "<br>";

text += animals[3] + "<br>";

text += animals[4] + "<br>";

text += animals[5] + "<br>";

Copy

### আপনি এই কোড লিখতে পারেনঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var animals = ["Tiger", "Lion", "Horse", "Hen"];

var text = "";

var i;

for (i = 0; i < animals.length; i++) {

text += animals[i] + "<br>";

}

document.getElementById("test").innerHTML = text;

</script>

</body>

</html>

Copy

### ফলাফল

Tiger

Lion

Horse

Hen

## লুপের ধরণ

জাভাস্ক্রিপ্টে বিভিন্ন ধরনের লুপ সাপোর্ট করেঃ

* **for** - একগুচ্ছ কোডকে একাধিকবার রান করানো যায়
* **for/in** - অবজেক্টের প্রোপার্টির মধ্যে লুপ করে
* **while** - যতক্ষন পর্যন্ত একটি নির্দেশিত শর্ত true হবে ততক্ষন পর্যন্ত কোড রান হবে
* **do/while** - এটিও while লুপের মত একটি নির্দেশিত শর্ত যতক্ষন true হবে ততক্ষন পর্যন্ত কোড রান হবে

## For লুপ

লুপ ব্যবহার করতে চাইলে আপনি প্রথমেই একটি for লুপ ব্যবহার করার চিন্তা করবেন।

for লুপের গঠনপ্রণালীঃ

for (statement 1; statement 2; statement 3) {

কোড

}

Copy

**statement 1** লুপ(কোড এক্সিকিউট হওয়ার) শুরু হওয়ার পূর্বে এক্সিকিউট হয়।

**statement 2** লুপের(কোড এক্সিকিউট হওয়ার) শর্ত নির্ধারন করে।

**statement 3** প্রতিবার লুপ(কোড এক্সিকিউট হওয়ার) এক্সিকিউট হওয়ার পর ইহা এক্সিকিউট হয়।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>বাটনে ক্লিক করলে ব্লকের কোডগুলো 4 বার লুপ হবে ।</h3>

<button onclick="myFunc()"> চেষ্টা করি </button>

<p id="test"></p>

<script>

function myFunc() {

var num = "";

var i;

for (i = 0; i < 4; i++) {

num += "এটা হলো নাম্বার " + i + "<br>";

}

document.getElementById("test").innerHTML = num;

}

</script>

</body>

</html>

Copy

### ফলাফল

### বাটনে ক্লিক করলে ব্লকের কোডগুলো 4 বার লুপ হবে ।

এটা হলো নাম্বার 0  
এটা হলো নাম্বার 1  
এটা হলো নাম্বার 2  
এটা হলো নাম্বার 3

উপরোক্ত উদাহরণ থেকেঃ

লুপ শুরু হওয়ার পূর্বে statement 1 একটি ভ্যারিয়েবল(var i = 0) সেট করে।

লুপের জন্য statement 2 শর্ত(i অবশ্যই 4 থেকে ছোট হতে হবে) নির্ধারন করে।

প্রতিবার লুপ এক্সিকিউট হওয়ার পর statement 3(i ++) এক করে মান বৃদ্ধি করে।

## statement 1

সাধারণত আপনি লুপের মধ্যে ব্যবহৃত ভ্যারিয়েবলকে(i = 0) ডিক্লেয়ার করার জন্য statement 1 ব্যবহার করবেন।

জাভাস্ক্রিপ্টে এটা গুরুত্বপূর্ন না। এখানে statement 1 ঐচ্ছিক।

আপনি statement 1 এর মধ্যে (কমা দিয়ে আলাদা করে ) অনেকগুলো ভ্যালু ব্যবহার করতে পারেনঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var animals = ["Tiger", "Lion", "Horse", "Hen"];

var i, len, text;

for (i = 0, len = animals.length, text = ""; i < len; i++) {

text += animals[i] + "<br>";

}

document.getElementById("test").innerHTML = text;

</script>

</body>

</html>

Copy

### ফলাফল

Tiger

Lion

Horse

Hen

এবং আপনি statement 1 কে না দিলেও পারেন(যদি তা আগেই ডিক্লেয়ার করা থাকে)।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var animals = ["Tiger", "Lion", "Horse", "Hen"];

var i = 2;

var len = animals.length;

var text = "";

for (; i < len; i++) {

text += animals[i] + "<br>";

}

document.getElementById("test").innerHTML = text;

</script>

</body>

</html>

Copy

### ফলাফল

Horse

Hen

statement 2

statement 2 সাধারণত ইনিশিয়াল ভ্যারিয়েবলের শর্ত যাচাই করার জন্য ব্যবহার করা হয়।

জাভাস্ক্রিপ্টে statement 2 ঐচ্ছিক।

statement 2 যদি true হয় তবে লুপ আবার শুরু হবে কিন্তু যদি এটি false হয় তবে লুপ এখানেই শেষ হয়ে যাবে।

যদি আপনি statement 2 না দেন তবে আপনাকে অবশ্যই লুপের ভিতর **break** দিতে হবে। অন্যথায় লুপ কখনোই শেষ হবে না এবং আপনার ব্রাউজার বন্ধ হয়ে যাবে। পরবর্তী পরিচ্ছেদে আপনি Break সম্পর্কে আরো শিখতে পারবেন।

## statement 3

সাধারণত statement 3 ইনিশিয়াল ভ্যারিয়েবলের মান বৃদ্ধি করে।

জাভাস্ক্রিপ্টে statement 3 ও ঐচ্ছিক।

statement 3 যেকোন কিছু করতে পারে যেমন ইনক্রিমেন্ট(i = i + 15), ডিক্রিমেন্ট(i--) অথবা অন্যকিছু।

statement 3 বাদ দেওয়া যেতে পারে(যদি আপনি লুপের ভিতর ভ্যালুকে ইনক্রিমেন্ট করেন)

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var animals = ["Tiger", "Lion", "Horse", "Hen"];

var i = 0;

var len = animals.length;

var text = "";

for (; i < len; ) {

text += animals[i] + "<br>";

i++;

}

document.getElementById("test").innerHTML = text;

</script>

</body>

</html>

Copy

### ফলাফল

Tiger

Lion

Horse

Hen

## For/In লুপ

জাভাস্ক্রিপ্টে for/in স্টেটমেন্ট একটি অবজেক্টের প্রোপার্টির মধ্যে লুপ করার জন্য ব্যবহার করা হয়ঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var text = "";

var person = {fname:"আজিজুর", lname:"রহমান ", age:32};

var a;

for (a in person) {

text += person[a] + " ";

}

document.getElementById("test").innerHTML = text;</script>

</body>

</html>

Copy

### ফলাফল

আজিজুর রহমান 32

# জাভাস্ক্রিপ্ট while লুপ

লুপ হচ্ছে একগুচ্ছ কোড যা নির্ধারিত শর্ত যতক্ষণ true থাকবে ততক্ষণ এক্সিকিউট হবে।

## while লুপ

while লুপ ততক্ষণ পর্যন্ত লুপ করতে থাকবে যতক্ষণ পর্যন্ত তার নির্ধারিত শর্ত true থাকে।

### গঠনপ্রণালী

while (condition) {

কোড

}

Copy

এখানে যতক্ষণ পর্যন্ত ভ্যারিয়েবল i এর মান 7 থেকে ছোট হবে ততক্ষণ কোডটি রান হবেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> বাটনে ক্লিক করলে ব্লকের কোডটি লুপ হতে থাকবে যতক্ষন পর্যন্ত i এর মান 7 থেকে ছোট থাকে ।</h3>

<button onclick="myFunc()"> চেষ্টা করি </button>

<p id="test"></p>

<script>

function myFunc() {

var num = "";

var i = 0;

while (i < 7) {

num += "<br> এটা হলো নাম্বার " + i;

i++;

}

document.getElementById("test").innerHTML = num;

}

</script>

</body>

</html>

Copy

### ফলাফল

### বাটনে ক্লিক করলে ব্লকের কোডটি লুপ হতে থাকবে যতক্ষন পর্যন্ত i এর মান 7 থেকে ছোট থাকে ।

এটা হলো নাম্বার 0  
এটা হলো নাম্বার 1  
এটা হলো নাম্বার 2  
এটা হলো নাম্বার 3  
এটা হলো নাম্বার 4  
এটা হলো নাম্বার 5  
এটা হলো নাম্বার 6

যদি আপনি ভ্যারিয়েবল i এর মান বৃদ্ধি না করেন তাহলে আপনার ব্রাউজার বন্ধ হয়ে যাবে কারণ লুপটি কখনোই থামবে না।

## do/while লুপ

do/while লুপ হচ্ছে while লুপের ভিন্ন রূপ। এটি শর্ত যাচাই করার পূর্বে কোড একবার রান করে এবং শর্তটি যতক্ষন পর্যন্ত true থাকে ততক্ষণ পর্যন্ত রান করে।

### গঠনপ্রণালী

do {

কোড

}

while (condition);

Copy

নিচের উদাহরণে do/while লুপ ব্যবহার করা হয়েছে। এখানে শর্ত যাচাইয়ের পূর্বে কোডকে অন্তত একবার রান করানো হয়ঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>বাটনে ক্লিক করলে ব্লকের কোডটি লুপ হতে থাকবে যতক্ষন পর্যন্ত i এর মান 7 থেকে ছোট থাকে ।</h3>

<button onclick="myFunc()">চেষ্টা কর</button>

<p id="test"></p>

<script>

function myFunc() {

var num = ""

var i = 0;

do {

num += "<br>এটা হলো নাম্বার " + i;

i++;

}

while (i < 7)

document.getElementById("test").innerHTML = num;

}

</script>

</body>

Copy

### ফলাফল

### বাটনে ক্লিক করলে ব্লকের কোডটি লুপ হতে থাকবে যতক্ষন পর্যন্ত i এর মান 7 থেকে ছোট থাকে ।

এটা হলো নাম্বার 0  
এটা হলো নাম্বার 1  
এটা হলো নাম্বার 2  
এটা হলো নাম্বার 3  
এটা হলো নাম্বার 4  
এটা হলো নাম্বার 5  
এটা হলো নাম্বার 6

ভ্যারিয়েবলের ভ্যালু বৃদ্ধি করতে ভুলে যাবেন না,নয়তো লুপ কখনোই বন্ধ হবে না।

## for এবং while এর মধ্যে সাদৃশ্যতা

পূর্ববর্তী পরিচ্ছেদে আপনি for লুপ সম্পর্কে পড়েছেন এবং বুঝতে পারতেছে for এবং while লুপ একই রকম যদি আমরা for লুপের statement 1 এবং statement 2 না দেই।

এই উদাহরণে animals অ্যারে থেকে প্রানীর নাম নেওয়ার জন্য আমরা **for** লুপ ব্যবহার করেছিঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var animals = ["Tiger", "Lion", "Horse", "Hen"];

var i = 0;

var text = "";

for (animals[i]) {

text += animals[i] + "<br>";

i++;

}

document.getElementById("test").innerHTML = text;

</script>

</body>

</html>

Copy

### ফলাফল

Tiger

Lion

Horse

Hen

এই উদাহরণে animals অ্যারে থেকে প্রানীর নাম নেওয়ার জন্য আমরা **while** লুপ ব্যবহার করেছিঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var animals = ["Tiger", "Lion", "Horse", "Hen"];

var i = 0;

var text = "";

while (animals[i]) {

text += animals[i] + "<br>";

i++;

}

document.getElementById("test").innerHTML = text;

</script>

</body>

</html>

Copy

### ফলাফল

Tiger

Lion

Horse

Hen

# জাভাস্ক্রিপ্ট break এবং continue

## Break স্টেটমেন্ট

এই টিউটোরিয়ালের পূর্ববর্তী পরিচ্ছেদে break স্টেটমেন্টের ব্যবহার দেখানো হয়েছিল। switch স্টেটমেন্ট থেকে বের হয়ে যাওয়ার জন্য ইহা ব্যবহার করা হয়েছিল।

একটি লুপ থেকে বের হয়ে যাওয়ার জন্যও break স্টেটমেন্ট ব্যবহার করা হয়।

break স্টেটমেন্ট লুপকে বন্ধ করে দেয় এবং পরবর্তী কোন কোড থাকলে তা এক্সিকিউট করেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h2>Break স্টেটমেন্টসহ একটি লুপ</h2>

<p id="test"></p>

<script>

var num = "";

var i;

for (i = 0; i < 7; i++) {

if (i === 4) { break; }

num += "সংখ্যাটি হলো " + i + "<br>";

}

document.getElementById("test").innerHTML = num;

</script>

</body>

</html>

Copy

### ফলাফল

## Break স্টেটমেন্টসহ একটি লুপ

সংখ্যাটি হলো 0

সংখ্যাটি হলো 1

সংখ্যাটি হলো 2

সংখ্যাটি হলো 3

## continue স্টেটমেন্ট

যদি একটি নির্দিষ্ট শর্ত ঘটে তাহলে continue স্টেটমেন্ট লুপের ঐ ধাপকে বাদ দিয়ে পরবর্তী ধাপে চলে যায়।

এই উদাহরণটিতে 4 এর মান এড়িয়ে যায়ঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h2> যে ধাপে i = 4 হবে সে ধাপে লুপ বন্ধ ( skip ) হবে ।</h2>

<p id="test"></p>

<script>

var num = "";

var i;

for (i = 0; i < 8; i++) {

if (i === 4) { continue; }

num += " সংখ্যাটি হলো " + i + "<br>";

}

document.getElementById("test").innerHTML = num;

</script>

</body>

</html>

Copy

### ফলাফল

## যে ধাপে i = 4 হবে সে ধাপে লুপ বন্ধ ( skip ) হবে ।

সংখ্যাটি হলো 0

সংখ্যাটি হলো 1

সংখ্যাটি হলো 2

সংখ্যাটি হলো 3

সংখ্যাটি হলো 5

সংখ্যাটি হলো 6

সংখ্যাটি হলো 7

## জাভাস্ক্রিপ্ট লেবেল

জাভাস্ক্রিপ্ট স্টেটমেন্টকে নাম দেওয়ার জন্য স্টেটমেন্টের পূর্বে লেবেলের নাম এবং কোলন ব্যবহার করতে হবেঃ

label:

statements

Copy

জাভাস্ক্রিপ্টে break ও continue স্টেটমেন্ট শুধুমাত্র জাভাস্ক্রিপ্ট কোডের ব্লক থেকে বের হয়ে আসতে পারে।

গঠনপ্রণালীঃ

break labelname;

continue labelname;

Copy

continue স্টেটমেন্টের(লেবেলসহ অথবা ছাড়া) মাধ্যমে লুপের একটি ধাপকে এড়িয়ে যাওয়ার জন্য ব্যবহার করা হয়।

একটি লুপ বা switch স্টেটমেন্ট থেকে বের হয়ে আসার জন্য লেবেল রেফারেন্স ছাড়া একটি break স্টেটমেন্ট ব্যবহার করা হয়।

যে কোন কোডের ব্লক থেকে বের হয়ে যাওয়ার জন্য একটি লেবেল রেফারেন্স দিয়ে break স্টেটমেন্ট ব্যবহার করা হয়ঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var animals = ["Tiger", "Lion", "Horse", "Hen"];

var text = "";

animalsList: {

text += animals[0] + "<br>";

text += animals[1] + "<br>";

text += animals[2] + "<br>";

break animalsList;

text += animals[3] + "<br>";

}

document.getElementById("test").innerHTML = text;

</script>

</body>

</html>

Copy

### ফলাফল

Tiger

Lion

Horse

# জাভাস্ক্রিপ্ট স্ট্রিং

টেক্সট সংরক্ষন এবং সেগুলো নিয়ে কাজ করার জন্য জাভাস্ক্রিপ্টে স্ট্রিং ব্যবহার করা হয়।

## জাভাস্ক্রিপ্ট স্ট্রিং

জাভাস্ক্রিপ্টে স্ট্রিং সাধারণত কিছু ক্যারেক্টারকে জমা রাখে।যেমন "Tamjid Hasan"।

একটি স্ট্রিং হলো উদ্ধৃতি চিহ্নের মাঝে যেকোন ধরনের টেক্সট। আপনি সিঙ্গেল বা ডাবল কোটেশন(উদ্ধৃতি চিহ্ন) ব্যবহার করতে পারেনঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var animal1 = "Tiger";

var animal2 = "Lion";

document.getElementById("test").innerHTML =

animal1 + "<br>" + animal2;

</script>

</body>

</html>

Copy

### ফলাফল

Tiger  
Lion

আপনি একটি স্ট্রিং ভিতরে উদ্ধৃতি চিহ্ন ব্যবহার করতে পারেন কিন্তু লক্ষ্য রাখতে হবে তার পার্শ্ববর্তী স্ট্রিংয়ের উদ্ধৃতি চিহ্নের সাথে না মিলেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var wel1 = "Welcome To Satt Academy"

var wel2 = "Welcome To 'Satt' Academy"

var wel3 = 'Welcome To "Satt" Academy'

document.getElementById("test").innerHTML =

wel1 + "<br>" + wel2 + "<br>" + wel3;

</script>

</body>

</html>

Copy

### ফলাফল

Welcome To Satt Academy  
Welcome To 'Satt' Academy  
Welcome To "Satt" Academy

## স্ট্রিংয়ের দৈর্ঘ্য(length)

একটি স্ট্রিংয়ের দৈর্ঘ্য স্ট্রিংয়ের বিল্ট-ইন **length** প্রোপার্টির মাধ্যমে পাওয়া যায়ঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var text = "WelcomeToSattAcademy";

document.getElementById("test").innerHTML = text.length;

</script>

</body>

</html>

Copy

### ফলাফল

20

## স্পেশাল ক্যারেক্টার

যেহেতু স্ট্রিংকে উদ্ধৃতির মধ্যে লেখা আবশ্যক, তাই জাভাস্ক্রিপ্ট নিচের স্ট্রিংকে ভুলভাবে পড়বেঃ

var a = "Welcome To "Satt" Academy"

Copy

স্ট্রিংটি "Welcome To" পযর্ন্ত নেওয়া হবে।

এই সমস্যা এড়ানোর জন্য আমরা **\** চিহ্নটি ব্যবহার করবো।

স্ট্রিংয়ের মধ্যে বিশেষ অক্ষর ব্যবহার করতে চাইলে আপনাকে ব্যাকস্ল্যাশ(\) ব্যবহার করতে হবেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var a = "Welcome To \"Satt\" Academy"

document.getElementById("test").innerHTML =

a;

</script>

</body>

</html>

Copy

### ফলাফল

Welcome To "Satt" Academy

স্ট্রিংয়ের মধ্যে অন্যান্য স্পেশাল ক্যারেক্টার ব্যবহার করতেও \ ব্যবহার করা হয়।

টেক্সট স্ট্রিং-এ ব্যাকস্ল্যাশ(backslash) ব্যবহার করে নিচের স্পেশাল ক্যারেক্টারগুলো তৈরি করা যায়ঃ

|  |  |
| --- | --- |
| **কোড** | **ফলাফল** |
| \' | সিঙ্গেল কোটেশন |
| \" | ডাবল কোটেশন |
| \\ | ব্যাকস্ল্যাশ |
| \n | নতুন লাইন |
| \r | নতুন লাইন তৈরি করার মতই |
| \t | ট্যাব |
| \b | ব্যাকস্পেস |
| \f | ফর্ম ফিড |

## লম্বা লাইনের কোড এড়িয়ে চলা

পড়ার সুবিধার জন্য প্রোগ্রামাররা ৮০ ক্যারেক্টারের বড় লাইনকে এড়িয়ে চলে।

যদি একটি জাভাস্ক্রিপ্ট স্টেটমেন্ট এক লাইনে সম্পন্ন না হয় তবে অপারেটরের পরে পরবর্তী লাইনে চলে যাওয়া সবচেয়ে ভালো পদ্ধতিঃ

document.getElementById("test").innerHTML =

"স্যাট সুহিন";

Copy

এছাড়াও আপনি কোন স্ট্রিংয়ের টেক্সটের মাঝে একটি ব্যাকস্ল্যাশ ব্যবহার করে লাইন ব্রেক করতে পারেনঃ

document.getElementById("demo").innerHTML = "স্যাট \

সুহিন";

Copy

ব্যাকস্ল্যাশ মেথডটি ইকমাস্ক্রিপ্ট(জাভাস্ক্রিপ্ট) স্ট্যান্ডার্ড নয়।  
কিছু ব্রাউজারে ব্যাকস্ল্যাশের(\) পরে স্পেস সাপোর্ট করে না।

স্ট্রিংকে ভাঙ্গার আরো একটি নিরাপদ পদ্ধতি হলো স্ট্রিংয়ের যোগ করাঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> স্ট্রিং ছাড়াও একটি স্ট্রিং-এর মধ্যে বিরতির জন্য একটি কোড লাইন ব্যবহার করা হয়।

</h4>

<p id="test"></p>

<script>

document.getElementById("test").innerHTML = "স্যাট " +

"সুহিন";

</script>

</body>

</html>

Copy

### ফলাফল

#### স্ট্রিং ছাড়াও একটি স্ট্রিং-এর মধ্যে বিরতির জন্য একটি কোড লাইন ব্যবহার করা হয়।

স্যাট সুহিন

আপনি ব্যাকস্ল্যাশ ব্যবহার করে কোডের মধ্যে লাইন ব্রেক করতে পারবেন নাঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> আপনি একটি ব্যাকস্ল্যাশ (\) দিয়ে একটি কোড লাইন বিরতি আনতে পারবেন না। </h4>

<script>

document.getElementById("test").innerHTML = \

"স্যাট সুহিন";

</script>

</body>

</html>

Copy

### ফলাফল

#### আপনি একটি ব্যাকস্ল্যাশ (\) দিয়ে একটি কোড লাইন বিরতি আনতে পারবেন না।

## স্ট্রিং অবজেক্ট

সাধারণত জাভাস্ক্রিপ্ট স্ট্রিং হলো প্রিমিটিভ ভ্যালু যা আক্ষরিকভাবে তৈরি করা হয়ঃ **var a = "Tamim"**

কিন্তু new কিওয়ার্ড দ্বারা স্ট্রিংকে অবজেক্টে হিসেবেও ডিফাইন করা যায়ঃ**var b = new String("Tamim")**

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var a = "Tamim";

var b = new String("Tamim");

document.getElementById("test").innerHTML = typeof a + "<br>" + typeof b;

</script>

</body>

</html>

Copy

### ফলাফল

string  
object

স্ট্রিংকে অবজেক্ট হিসাবে তৈরি করবেন না। এটি এক্সিকিউশনের গতিকে ধীর করে দেবে।  
**new** কিওয়ার্ড কোডকে জটিল করে। ইহা অপ্রত্যাশিত ফলাফল দেখাতে পারেঃ

যখন আমরা == অপারেটরটি ব্যবহার করবো,দুইটি স্ট্রিং সমান দেখায়ঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> স্ট্রিং-কে অবজেক্টের মত তৈরি করবেন না। </h4>

<h4>স্ট্রিং এবং অবজেক্টের সাথে কখনো তুলনা করা ঠিক নয়।</h4>

<p id="test"></p>

<script>

var a = "Tamim";

var b = new String("Tamim");

document.getElementById("test").innerHTML = (a==b);

</script>

</body>

</html>

Copy

### ফলাফল

#### স্ট্রিং-কে অবজেক্টের মত তৈরি করবেন না।

#### স্ট্রিং এবং অবজেক্টের সাথে কখনো তুলনা করা ঠিক নয়।

true

যখন আমরা === অপারেটরটি ব্যবহার করবো, দুইটি স্ট্রিং সমান হবে না কারন === অপারেটরের ক্ষেত্রে মান এবং টাইপ উভয়ই একই হতে হবে।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> স্ট্রিং-কে অবজেক্টের মত তৈরি করবেন না। </h4>

<h4>স্ট্রিং এবং অবজেক্টের সাথে কখনো তুলনা করা ঠিক নয়।</h4>

<p id="test"></p>

<script>

var a = "Tamim";

var b = new String("Tamim");

document.getElementById("test").innerHTML = (a===b);

</script>

</body>

</html>

Copy

### ফলাফল

#### স্ট্রিং-কে অবজেক্টের মত তৈরি করবেন না।

#### স্ট্রিং এবং অবজেক্টের সাথে কখনো তুলনা করা ঠিক নয়।

False

অবজেক্টকে তুলনা করা যাবে নাঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> স্ট্রিং-কে অবজেক্টের মত তৈরি করবেন না। </h4>

<h4>স্ট্রিং এবং অবজেক্টের সাথে কখনো তুলনা করা ঠিক নয়।</h4>

<p id="test"></p>

<script>

var a = new String("Tamim");

var b = new String("Tamim");

document.getElementById("test").innerHTML = (a==b);

</script>

</body>

</html>

Copy

### ফলাফল

#### স্ট্রিং-কে অবজেক্টের মত তৈরি করবেন না।

#### স্ট্রিং এবং অবজেক্টের সাথে কখনো তুলনা করা ঠিক নয়।

false

# জাভাস্ক্রিপ্ট স্ট্রিং মেথড

স্ট্রিং মেথড আপনাকে স্ট্রিং নিয়ে কাজ করতে সাহায্য করবে।

## স্ট্রিং মেথড এবং প্রোপার্টি

প্রিমিটিভ(Primitive) ভ্যালুর, যেমন "Tamjid Hasan", কোন প্রোপার্টি বা মেথড থাকতে পারে না(কারন তারা অবজেক্ট নয়)।

## স্ট্রিং লেন্থ

**length** প্রোপার্টি স্ট্রিংটি কয়টি ক্যারেক্টার নিয়ে গঠিত তা রিটার্ন করেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var text = "WelcomeToSattAcademy";

document.getElementById("test").innerHTML = text.length;

</script>

</body>

</html>

Copy

### ফলাফল

20

## স্ট্রিংয়ের মধ্যে স্ট্রিং খোঁজা

**indexOf()** মেথডটি একটি স্ট্রিংয়ের মধ্যে একটি নির্দিষ্ট টেক্সটের **প্রথম** অবস্থান রিটার্ন করেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4 id="para">Please locate where 'locate' occurs!</h4>

<button onclick="myFunc()"> নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

function myFunc() {

var str = document.getElementById("para").innerHTML;

var position = str.indexOf("locate");

document.getElementById("test").innerHTML = position;

}

</script>

</body>

</html>

Copy

### ফলাফল

#### Please locate where 'locate' occurs!

7

**lastIndexOf()** মেথডটি একটি স্ট্রিংয়ের মধ্যে একটি নির্দিষ্ট টেক্সটের **শেষ** অবস্থান রিটার্ন করেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4 id="para">Please locate where 'locate' occurs!</h4>

<button onclick="myFunc()"> নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

function myFunc() {

var str = document.getElementById("para").innerHTML;

var position = str.lastIndexOf("locate");

document.getElementById("test").innerHTML = position;

}

</script>

</body>

</html>

Copy

### ফলাফল

#### Please locate where 'locate' occurs!

21

যদি টেক্সটটি না পাওয়া যায় indexOf() এবং lastIndexOf() উভয় মেথডই -1 রিটার্ন করে।

জাভাস্ক্রিপ্টে স্ট্রিংয়ের অবস্থান শূন্য থেকে গণনা করা হয়।  
স্ট্রিং-এ ০(শূন্য) হচ্ছে প্রথম অবস্থান, ১ হচ্ছে দ্বিতীয় অবস্থান আর ২ হচ্ছে তৃতীয় অবস্থান এবং এইভাবে চলতে থাকে।

উভয় মেথডই দ্বিতীয় একটি প্যারামিটার গ্রহণ করে।

## স্ট্রিংয়ের মধ্যে একটি স্ট্রিং অনুসন্ধান

**search()** মেথডটি একটি স্ট্রিংয়ের মধ্যে একটি নির্দিষ্ট ভ্যালু অনুসন্ধান করে এবং অনুসন্ধানকৃত ভ্যালুর অবস্থান রিটার্ন করেনঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4 id="para">Please locate where 'locate' occurs!</h4>

<button onclick="myFunc()"> নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

function myFunc() {

var str = document.getElementById("para").innerHTML;

var position = str.search("locate");

document.getElementById("test").innerHTML = position;

}

</script>

</body>

</html>

Copy

### ফলাফল

#### Please locate where 'locate' occurs!

7

## আপনি লক্ষ্য করেছিলেন কি?

indexOf() এবং search() উভয় মেথড একই।

তারা উভয়ই একই আর্গুমেন্ট(arguments) বা প্যারামিটার ভ্যালু গ্রহন করে এবং একই মান রিটার্ন করে।

দুটি মেথড একই, কিন্তু search() মেথডটি অনেক বেশি শক্তিশালী।

আপনি অনুসন্ধান সম্পর্কে রেগুলার এক্সপ্রেশন(regular expressions) পরিচ্ছেদে আরো জানবেন।

## স্ট্রিংকে বিভক্ত করা

স্ট্রিংয়ের একটি অংশকে বিভক্ত করার জন্য ৩টি মেথড রয়েছেঃ

* slice(start, end)
* substring(start, end)
* substr(start, length)

## slice() মেথড

**slice()** মেথডটি স্ট্রিংয়ের একটি অংশকে ভাগ করে এবং বিভক্ত অংশটিকে নতুন একটি স্ট্রিংয়ে রিটার্ন করে।

মেথডটি দুইটি প্যারামিটার গ্রহন করেঃ শুরুর অবস্থান(start index) এবং শেষের অবস্থান(end index)।

এই উদাহরণে একটি স্ট্রিংয়ের ৭ থেকে ১৩ অবস্থানের অংশটুকু ভাগ করে নেয়ঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> slice() মেথড স্ট্রিং-এর একটি অংশকে এক্সট্রাক্ট করে এবং এক্সট্রাক্ট করা অংশকে নতুন স্ট্রিং-এ রিটার্ন করে। </h4>

<p id="test"></p>

<script>

var animals = "Tiger, Lion, Elephant";

document.getElementById("test").innerHTML = animals.slice(7,11);

</script>

</body>

</html>

Copy

### ফলাফল

#### slice() মেথড স্ট্রিং-এর একটি অংশকে এক্সট্রাক্ট করে এবং এক্সট্রাক্ট করা অংশকে নতুন স্ট্রিং-এ রিটার্ন করে।

Lion

যদি কোন প্যারামিটারের মান ঋনাত্মক হয় তবে স্ট্রিংয়ের অবস্থান গণনা শেষ থেকে শুরু হবে।

এই উদাহরণে স্ট্রিংয়ের -১২ থেকে -৬ অংশটি ভাগ করা হবে। যেহেতু প্যারামিটারের মান ঋনাত্মক এটি স্ট্রিংয়ের শেষ থেকে গণনা শুরু করবে।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>slice() মেথড স্ট্রিং-এর একটি অংশকে এক্সট্রাক্ট করে এবং এক্সট্রাক্ট করা অংশকে নতুন স্ট্রিং-এ রিটার্ন করে। </h4>

<p id="test"></p>

<script>

var animals = "Tiger, Lion, Elephant";

document.getElementById("test").innerHTML = animals.slice(-15, -10);

</script>

</body>

</html>

Copy

### ফলাফল

#### slice() মেথড স্ট্রিং-এর একটি অংশকে এক্সট্রাক্ট করে এবং এক্সট্রাক্ট করা অংশকে নতুন স্ট্রিং-এ রিটার্ন করে।

Lion

আপনি যদি ২য় প্যারামিটারটি উল্লেখ না করেন তাহলে এটি প্রথম অবস্থান থেকে বাকী সম্পূর্ন অংশ ভাগ করবেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> slice() মেথড স্ট্রিং-এর একটি অংশকে এক্সট্রাক্ট করে এবং এক্সট্রাক্ট করা অংশকে নতুন স্ট্রিং-এ রিটার্ন করে। </h4>

<p id="test"></p>

<script>

var animals = "Tiger, Lion, Elephant";

document.getElementById("test").innerHTML = animals.slice(13);

</script>

</body>

</html>

Copy

### ফলাফল

#### slice() মেথড স্ট্রিং-এর একটি অংশকে এক্সট্রাক্ট করে এবং এক্সট্রাক্ট করা অংশকে নতুন স্ট্রিং-এ রিটার্ন করে।

Elephant

অথবা এটি শেষ থেকে গণনা করবেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> slice() মেথড স্ট্রিং-এর একটি অংশকে এক্সট্রাক্ট করে এবং এক্সট্রাক্ট করা অংশকে নতুন স্ট্রিং-এ রিটার্ন করে। </h4>

<p id="test"></p>

<script>

var animals = "Tiger, Lion, Elephant";

document.getElementById("test").innerHTML = animals.slice(-15);

</script>

</body>

</html>

Copy

### ফলাফল

#### slice() মেথড স্ট্রিং-এর একটি অংশকে এক্সট্রাক্ট করে এবং এক্সট্রাক্ট করা অংশকে নতুন স্ট্রিং-এ রিটার্ন করে।

Lion, Elephant

ইন্টারনেট এক্সপ্লোরার ৮ এবং এর আগের ব্রাউজারগুলোতে প্যারামিটারের ঋনাত্মক মান কাজ করে না।

## substring() মেথড

**substring()** মেথডটি slice() মেথডের মতই।

পার্থক্য হচ্ছে substring() মেথড ঋনাত্মক ইন্ডেক্স ভ্যালু গ্রহণ করে না।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> substring() মেথড স্ট্রিং-এর একটি অংশকে এক্সট্রাক্ট করে এবং এক্সট্রাক্ট করা অংশকে নতুন স্ট্রিং-এ রিটার্ন করে। </h4>

<p id="test"></p>

<script>

var animals = "Tiger, Lion, Elephant";

document.getElementById("test").innerHTML = animals.substring(7,11);

</script>

</body>

</html>

Copy

### ফলাফল

#### ubstring() মেথড স্ট্রিং-এর একটি অংশকে এক্সট্রাক্ট করে এবং এক্সট্রাক্ট করা অংশকে নতুন স্ট্রিং-এ রিটার্ন করে।

Lion

আপনি যদি ২য় প্যারামিটারটি উল্লেখ না করেন তাহলে substring() মেথড প্রথম অবস্থান থেকে বাকী সম্পূর্ন অংশ ভাগ করবে।

## substr() মেথড

**substr()** মেথডটি slice() মেথডের মতই।

পার্থক্য হচ্ছে দ্বিতীয় প্যারামিটারটিতে কতটি ক্যারেক্টার ভাগ করতে চান তা উল্লেখ করতে হবে।

<

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> substr() মেথড স্ট্রিং-এর একটি অংশকে এক্সট্রাক্ট করে এবং এক্সট্রাক্ট করা অংশকে নতুন স্ট্রিং-এ রিটার্ন করে। </h4>

<p id="test"></p>

<script>

var animals = "Tiger, Lion, Elephant";

document.getElementById("test").innerHTML = animals.substr(7,4);

</script>

</body>

</html>

Copy

### ফলাফল

#### substr() মেথড স্ট্রিং-এর একটি অংশকে এক্সট্রাক্ট করে এবং এক্সট্রাক্ট করা অংশকে নতুন স্ট্রিং-এ রিটার্ন করে।

Lion

যদি প্রথম প্যারামিটারের মান ঋনাত্মক হয় তাহলে স্ট্রিংয়ের শেষ প্রান্ত থেকে গণনা শুরু হয়।

২য় প্যারামিটারের মান কথনোই ঋনাত্মক হতে পারবে না কারন এটি দৈর্ঘ্য নির্দেশ করে।

আপনি যদি ২য় প্যারামিটারটি উল্লেখ না করেন তাহলে substr() মেথড বাকী সম্পূর্ন অংশ ভাগ করবে।

## স্ট্রিংয়ের কন্টেন্ট প্রতিস্থাপন

**replace()** মেথডটি স্ট্রিংয়ের নির্দিষ্ট একটি মানকে অন্য একটি ভ্যালু দ্বারা প্রতিস্থাপন করেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> নিচের প্যারাগ্রাফে "Codeacademy" বদলে "sattacademy" প্রতিস্থাপন করুন </h4>

<button onclick="myFunc()">এটি নিজে চেষ্টা করি </button>

<p id="test"> দয়া করে Codeacademy পরিদর্শন করুন!</p>

<script>

function myFunc() {

var str = document.getElementById("test").innerHTML;

var text = str.replace("Codeacademy","sattacademy");

document.getElementById("test").innerHTML = text;

}

</script>

</body>

</html>

Copy

### ফলাফল

#### নিচের প্যারাগ্রাফে "Codeacademy" বদলে "sattacademy" প্রতিস্থাপন করুন

দয়া করে sattacademy পরিদর্শন করুন!

replace() মেথড অনুসন্ধান ভ্যালু হিসাবে রেগুলার এক্সপ্রেশনও নিতে পারে।

স্বাভাবিকভাবে replace() মেথডটি অনুসন্ধানকৃত অংশকে প্রথম যেখানে পাবে তাকেই প্রতিস্থাপন করবে। যদি সবগুলোকে প্রতিস্থাপন করতে চান সেক্ষেত্রে রেগুলার এক্সপ্রেশনের g ফ্ল্যাগ(গ্লোবাল সার্চের জন্য) ব্যবহার করতে হবেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> নিচের প্যারাগ্রাফে "Codeacademy" বদলে "sattacademy" প্রতিস্থাপন করুন </h4>

<button onclick="myFunc()">এটি নিজে চেষ্টা করি </button>

<p id="test"> দয়া করে Codeacademy এবং Codeacademy পরিদর্শন করুন!</p>

<script>

function myFunc() {

var str = document.getElementById("test").innerHTML;

var text = str.replace(/Codeacademy/g,"sattacademy");

document.getElementById("test").innerHTML = text;

}

</script>

</body>

</html>

Copy

### ফলাফল

#### নিচের প্যারাগ্রাফে "Codeacademy" বদলে "sattacademy" প্রতিস্থাপন করুন

দয়া করে sattacademy এবং sattacademy পরিদর্শন করুন!

replace() মেথডটি স্ট্রিংকে পরিবর্তন করে না।এটা নতুন একটি স্ট্রিং রিটার্ন করে।

## বড়হাতের অক্ষর(Upper Case) এবং ছোটহাতের অক্ষরে(Lower Case) রূপান্তর

একটি স্ট্রিংকে **toUpperCase()** মেথডের মাধ্যমে বড় হাতের অক্ষরে রূপান্তর করা হয়েছেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>স্ট্রিং-কে বড় হাতের অক্ষরে রুপান্তরঃ </h4>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test">Welcome To Satt Academy!</p>

<script>

function myFunc() {

var text = document.getElementById("test").innerHTML;

document.getElementById("test").innerHTML = text.toUpperCase();

}

</script>

</body>

</html>

Copy

### ফলাফল

#### স্ট্রিং-কে বড় হাতের অক্ষরে রুপান্তরঃ

WELCOME TO SATT ACADEMY!

একটি স্ট্রিংকে **toLowerCase()** মেথডের মাধ্যমে ছোটহাতের অক্ষরে রূপান্তর করা হয়েছেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>স্ট্রিং-কে ছোট হাতের অক্ষরে রুপান্তরঃ </h4>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test">Welcome To Satt Academy!</p>

<script>

function myFunc() {

var text = document.getElementById("test").innerHTML;

document.getElementById("test").innerHTML = text.toLowerCase();

}

</script>

</body>

</html>

Copy

### ফলাফল

#### স্ট্রিং-কে ছোট হাতের অক্ষরে রুপান্তরঃ

welcome to satt academy!

## concat() মেথড

স্ট্রিং যোগ করার জন্য **concat()** মেথড ব্যবহার করা হয়ঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>concat() মেথড দুই বা তার বেশি স্ট্রিং-কে সংযুক্ত করে </h4>

<p id="test"></p>

<script>

var word1 = "Satt";

var word2 = "Academy!";

document.getElementById("test").innerHTML = word1.concat(" ",word2);

</script>

</body>

</html>

Copy

### ফলাফল

#### concat() মেথড দুই বা তার বেশি স্ট্রিং-কে সংযুক্ত করে

Satt Academy!

**concat()** মেথডটি প্লাস(+) অপারেটরের পরিবর্তে ব্যবহার করা যেতে পারে। দুইটি লাইন একই কাজ করেঃ

var word = "Satt" + " " + "Academy!";

var word = "Satt".concat(" ","Academy!");

Copy

সকল স্ট্রিং মেথডই একটি নতুন স্ট্রিং রিটার্ন করে। তারা মূল স্ট্রিংকে পরিবর্তন করে না।  
স্ট্রিং অপরিবর্তনীয়ঃ স্ট্রিংগুলো পরিবর্তন করা যাবে না শুধুমাত্র প্রতিস্থাপন করা যায়।

## স্ট্রিংয়ের ক্যারেক্টার ভাগ

স্ট্রিং ক্যারেক্টারকে ভাগ করার দুইটি পদ্ধতি রয়েছেঃ

* charAt(position)
* charCodeAt(position)

## charAt() মেথড

**charAt()** মেথডটি একটি স্ট্রিংয়ের একটি উল্লেখিত অবস্থানের(index) ক্যারেক্টারকে রিটার্ন করেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> charAt () মেথড স্ট্রিং-এর দেওয়া কারেক্টার অনুযায়ী অবস্থান রিটার্ন করেঃ </h4>

<p id="test"></p>

<script>

var sentence = "SATT ACADEMY";

document.getElementById("test").innerHTML = sentence.charAt(0);

</script>

</body>

</html>

Copy

### ফলাফল

#### charAt () মেথড স্ট্রিং-এর দেওয়া কারেক্টার অনুযায়ী অবস্থান রিটার্ন করেঃ

S

## charCodeAt() মেথড

**charCodeAt()** মেথডটি একটি স্ট্রিংয়ের একটি উল্লেখিত অবস্থানের(index) ক্যারেক্টারের ইউনিকোড রিটার্ন করেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> charCodeAt () মেথড স্ট্রিং-এর দেওয়া কারেক্টারের ইউনিকোড অনুযায়ী অবস্থান রিটার্ন করেঃ </h4>

<p id="test"></p>

<script>

var sentence = "SATT ACADEMY";

document.getElementById("test").innerHTML = sentence.charCodeAt(1);

</script>

</body>

</html>

Copy

### ফলাফল

#### charCodeAt () মেথড স্ট্রিং-এর দেওয়া কারেক্টারের ইউনিকোড অনুযায়ী অবস্থান রিটার্ন করেঃ

65

## স্ট্রিংকে অ্যারের মত করে এক্সেস করা নিরাপদ নয়

আপনি হয়তো স্ট্রিংকে অ্যারের মত করে এক্সেস করা দেখতে পারেনঃ

var sentence = "SATT ACADEMY";

sentence[1]; // A রিটার্ন করে

Copy

ইহা বিপজ্জনক এবং অনিশ্চিত ফলাফল দেখাতে পারেঃ

* এটা সকল ব্রাউজারে কাজ করে না(IE5, IE6, IE7 এ কাজ করে না)
* এটা স্ট্রিংকে অ্যারের মত করে তোলে(কিন্তু তারা অ্যারে না)
* str[0] = "H" কোন ভুল দেখায় না(কিন্তু কাজও করে না)

যদি আপনি একটি স্ট্রিংকে অ্যারে আকারে পড়তে চান, প্রথমে একে অ্যারেতে রূপান্তর করুন।

## স্ট্রিংকে অ্যারেতে রূপান্তর

**split()** মেথড ব্যবহার করে একটি স্ট্রিংকে অ্যারেতে রূপান্তরিত করা যেতে পারেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>প্রথম অ্যারে এলিমেন্ট প্রদর্শনের জন্য "নিজে চেষ্টা করি" ক্লিক করুনঃ </h4>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

function myFunc() {

var char = "m,s,f,e,h,k";

var arr = char.split(",");

document.getElementById("test").innerHTML = arr[0];

}

</script>

</body>

</html>

Copy

### ফলাফল

#### প্রথম অ্যারে এলিমেন্ট প্রদর্শনের জন্য "নিজে চেষ্টা করি" ক্লিক করুনঃ

m

যদি বিভাজক বাদ দেওয়া হয়, তাহলে সম্পূর্ন স্ট্রিংটি অ্যারের ০ ইনডেক্সে রাখা হবে।

বিভাজক যদি "" হয়, তবে অ্যারের মধ্যে প্রতিটি ক্যারেক্টারকে ভিন্ন ইনডেক্সে রাখা হবেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>প্রথম দুটি অ্যারে এলিমেন্ট প্রদর্শনের জন্য "নিজে চেষ্টা করি" ক্লিক করুনঃ </h4>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

function myFunc() {

var word = "Satt";

var arr = word.split("");

document.getElementById("test").innerHTML = arr[0] + "<br>" + arr[1];

}

</script>

</body>

</html>

Copy

### ফলাফল

#### প্রথম দুটি অ্যারে এলিমেন্ট প্রদর্শনের জন্য "নিজে চেষ্টা করি" ক্লিক করুনঃ

S  
a

## সম্পূর্ণ স্ট্রিং রেফারেন্স

সম্পূর্ণ রেফারেন্স জন্য আমাদের [সম্পূর্ণ জাভাস্ক্রিপ্ট স্ট্রিং রেফারেন্সে](https://www.sattacademy.com/jsref/jsref_obj_string.php) যান।

রেফারেন্সে সকল স্ট্রিং প্রোপার্টি এবং মেথডের উদাহরণ রয়েছে।

# জাভাস্ক্রিপ্ট অ্যারে

[« পূর্ববর্তী](https://www.sattacademy.com/js/js_date_methods.php)

[পরবর্তী »](https://www.sattacademy.com/js/js_array_methods.php)

একটি ভ্যারিয়েবলের মধ্যে অনেক ভ্যালু রাখার জন্য অ্যারে ব্যবহার করা হয়।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals;

</script>

</body>

</html>

Copy

### ফলাফল

Lion,Tiger,Horse,Hen

## অ্যারে কি?

অ্যারে একটি বিশেষ ভ্যারিয়েবল যা একসঙ্গে একাধিক মান ধারণ করতে পারে।

আপনার কাছে যদি একাধিক ভ্যালুর একটি লিস্ট থাকে(উদাহরণস্বরুপঃ স্যাট মেম্বারদের নামের লিস্ট) তাহলে সেগুলোকে আলাদা আলাদা ভ্যারিয়েবলের মধ্যে রাখলে নিচের মত দেখাবেঃ

var satt1 = "Azizur";

var satt2 = "Shapla";

var satt3 = "Tamim";

var satt4 = "Tahmid";

Copy

আপনার কাছে যদি ৩০০ মেম্বারের লিস্ট থাকে এবং লুপের মাধ্যমে একজন মেম্বারকে পেতে চান!

এর সমাধান হচ্ছে অ্যারে।

অ্যারে একই নামের অধীনে অনেক ভ্যালু রাখতে পারে এবং আপনি ইন্ডেক্স নম্বর ব্যবহার করে ভ্যালু এক্সেস করতে পারেন।

## অ্যারে তৈরি

অ্যারে লিটারাল ব্যবহার জাভাস্ক্রিপ্ট অ্যারে তৈরি করার সবচেয়ে সহজ উপায়।

সিন্টেক্সঃ

var array-name = [item1, item2, ...];

Copy

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals;

</script>

</body>

</html>

Copy

### ফলাফল

Lion,Tiger,Horse,Hen

অ্যারে ডিক্লেয়ারেশন একাধিক লাইনেও করা যায়ঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var animals = [

"Lion",

"Tiger",

"Horse",

"Hen"

];

document.getElementById("test").innerHTML = animals;

</script>

</body>

</html>

Copy

### ফলাফল

Lion,Tiger,Horse,Hen

শেষের এলিমেন্টের পরে কখনো কমা ব্যবহার করবেন না (যেমনঃ "Hen",) ।

## জাভাস্ক্রিপ্ট new কিওয়ার্ড ব্যবহার

নিম্নলিখিত উদাহরণেও একটি অ্যারে তৈরি করা হয়েছেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var animals = new Array("Lion", "Tiger", "Horse", "Hen");

document.getElementById("test").innerHTML = animals;

</script>

</body>

</html>

Copy

### ফলাফল

Lion,Tiger,Horse,Hen

উপরে দুটি উদাহরণ একই কাজ করে। এখানে new array() ব্যবহার করার প্রয়োজন নেই।  
সহজ, পাঠযোগ্যতা এবং এক্সিকিউশন গতির জন্য প্রথম পদ্ধতিটি ব্যবহার করুন।

## অ্যারের এলিমেন্ট এক্সেস

আপনি ইন্ডেক্স নম্বর উল্লেখ করার মাধ্যমে একটি অ্যারে এলিমেন্ট নির্দেশ করতে পারেন। নিচের স্টেটমেন্টে satt অ্যারের তৃতীয় এলিমেন্টকে এক্সেস করা হয়েছেঃ

var name = animals[2];

Copy

এই স্টেটমেন্টে তৃতীয় এলিমেন্টকে পরিবর্তন করা হয়েছেঃ

animals[2] = "Elephant";

Copy

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals[1];

</script>

</body>

</html>

Copy

### ফলাফল

Tiger

একটি অ্যারের প্রথম এলিমেন্ট হচ্ছে [0], দ্বিতীয়টি হচ্ছে [1]। অ্যারের ইনডেক্স ০ দিয়ে শুরু হয়।

## সম্পূর্ণ অ্যারে অ্যাক্সেস করা

জাভাস্ক্রিপ্টে সম্পূর্ন অ্যারেকে এক্সেস করার জন্য শুধুমাত্র অ্যারের নাম লিখতে হয়ঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals;

</script>

</body>

</html>

Copy

### ফলাফল

Lion,Tiger,Horse,Hen

## অ্যারে হচ্ছে অবজেক্ট

অ্যারে স্পেশাল টাইপের অবজেক্ট। জাভাস্ক্রিপ্টের **typeof** অপারেটর অ্যারের ক্ষেত্রে object রিটার্ন করে।

কিন্তু, জাভাস্ক্রিপ্ট অ্যারেকে অ্যারে হিসেবে সবচেয়ে ভালো বর্ণনা করা যায়।

অ্যারে সংখ্যা ব্যবহার করে তার এলিমেন্টকে এক্সেস করে। এই উদাহরণে **animals[1]** দ্বারা অ্যারেকে এক্সেস করলে Tiger রিটার্ন করেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals[1];

</script>

</body>

</html>

Copy

### ফলাফল

Tiger

অবজেক্ট তার মেম্বারকে এক্সেস করার জন্য নাম ব্যবহার করে। এই উদাহরনে, **member.firstName** দ্বারা অবজেক্টকে এক্সেস করলে 'আজিজুর' রিটার্ন করেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h2>জাভাস্ক্রিপ্ট অবজেক্ট</h2>

<p>জাভাস্ক্রিপ্ট অবজেক্ট প্রোপার্টি অ্যাক্সেস করতে নাম ব্যবহার করে। </p>

<p id="test"></p>

<script>

var member = {firstName:"আজিজুর", lastName:"রহমান", age:32};

document.getElementById("test").innerHTML = member["firstName"];

</script>

</body>

</html>

Copy

### ফলাফল

## জাভাস্ক্রিপ্ট অবজেক্ট

জাভাস্ক্রিপ্ট অবজেক্ট প্রোপার্টি অ্যাক্সেস করতে নাম ব্যবহার করে।

আজিজুর

## অ্যারে এলিমেন্ট অবজেক্ট হতে পারে

জাভাস্ক্রিপ্ট ভ্যারিয়েবল অবজেক্ট হতে পারে। অ্যারে হচ্ছে স্পেশাল টাইপের অবজেক্ট।

এই কারনে, বিভিন্ন ধরনের ভ্যারিয়েবলকে একই অ্যারেতে রাখা যায়।

একটি অ্যারেতে আপনি অবজেক্ট, ফাংশন এবং অন্য একটি অ্যারেও রাখতে পারেনঃ

myArray[0] = Date.now;

myArray[1] = myFunc;

myArray[2] = mysatt;

Copy

## অ্যারে প্রোপার্টি এবং মেথড

জাভাস্ক্রিপ্ট অ্যারের সবচেয়ে শক্তিশালী দিক হচ্ছে এর বিল্ট-ইন প্রোপার্টি এবং মেথডঃ

var a = satt.length; // এলিমেন্টের সংখ্যা রিটার্ন করে

var b = satt.sort(); // sort() মেথড অ্যারেকে শ্রেণীবদ্ধ করে

Copy

অ্যারে মেথড সম্পর্কে পরবর্তী পরিচ্ছেদে আরো ভালোভাবে আলোচনা করা হয়েছে।

## length প্রোপার্টি

**length** প্রোপার্টি একটি অ্যারের এলিমেন্টের সংখ্যা রিটার্ন করে।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h2>জাভাস্ক্রিপ্ট অ্যারে</h2>

<p> length প্রোপার্টি একটি অ্যাারের length রিটার্ন করে। </p>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals.length;

</script>

</body>

</html>

Copy

### ফলাফল

## জাভাস্ক্রিপ্ট অ্যারে

length প্রোপার্টি একটি অ্যাারের length রিটার্ন করে।

4

## অ্যারে এলিমেন্ট যোগ করা

অ্যারেতে নতুন এলিমেন্ট যোগ করার সবচেয়ে সহজ পদ্ধতি হচ্ছে push মেথড ব্যবহার করাঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> push মেথড একটি অ্যারেতে নতুন এলিমেন্টকে সংযুক্ত করে। </h3>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals;

function myFunc() {

animals.push("Elephant");

document.getElementById("test").innerHTML = animals;

}

</script>

</body>

</html>

Copy

### ফলাফল

### push মেথড একটি অ্যারেতে নতুন এলিমেন্টকে সংযুক্ত করে।

Lion,Tiger,Horse,Hen,Elephant

length প্রোপার্টিও ব্যবহার করে অ্যারে তে নতুন প্রোপার্টি যোগ করা যায়ঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> length প্রোপার্টি push() মেথড ছাড়াই একটি অ্যাারেতে নতুন এলিমেন্ট সহজ পদ্ধতিতে সংযুক্ত করে। </h3>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals;

function myFunc() {

animals[animals.length] = "Elephant";

document.getElementById("test").innerHTML = animals;

}

</script>

</body>

</html>

Copy

### ফলাফল

### length প্রোপার্টি push() মেথড ছাড়াই একটি অ্যাারেতে নতুন এলিমেন্ট সহজ পদ্ধতিতে সংযুক্ত করে।

Lion,Tiger,Horse,Hen,Elephant

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> একটি অ্যারের মধ্যে উচ্চ ইনডেক্স সমূহের সঙ্গে এলিমেন্ট যোগ করে undefined তৈরি করতে পারেন। </h3>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals;

function myFunc() {

animals[9] = "Elephant";

document.getElementById("test").innerHTML = animals[6];

}

</script>

</body>

</html>

Copy

### ফলাফল

### একটি অ্যারের মধ্যে উচ্চ ইনডেক্স সমূহের সঙ্গে এলিমেন্ট যোগ করে undefined তৈরি করতে পারেন।

undefined

## অ্যারে এলিমেন্টে লুপিং করা

অ্যারে এলিমেন্টের মধ্যে লুপ করার জন্য একটি সবচেয়ে ভালো পদ্ধতি হচ্ছে "for" লুপ ব্যবহার করাঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>একটি অ্যারের মাধ্যমে লুপ করার সবচেয়ে ভালো উপায় লুপের জন্য একটি মান ব্যবহার করাঃ </h3>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];;

var aLength = animals.length;

var result;

var i;

result = "<p>";

for (i = 0; i < aLength; i++) {

result += animals[i] + "<br>";

}

result += "</p>";

document.getElementById("test").innerHTML = result;

</script>

</body>

</html>

Copy

### ফলাফল

### একটি অ্যারের মাধ্যমে লুপ করার সবচেয়ে ভালো উপায় লুপের জন্য একটি মান ব্যবহার করাঃ

Lion  
Tiger  
Horse  
Hen

## অ্যাসোসিয়েটিভ অ্যারে

অ্যারের ইনডেক্সিং নাম দিয়ে অনেক প্রোগ্রামিং ল্যাংগুয়েজে সাপোর্ট করে।

যেসকল অ্যারের ইনডেক্স নাম দিয়ে করা হয় তাদেরকে অ্যাসোসিয়েটিভ অ্যারে বলে।

জাভাস্ক্রিপ্টে এটা সাপোর্ট করে না।

জাভাস্ক্রিপ্টে, অ্যারে সবসময় নম্বর ইনডেক্স ব্যবহার করে।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var member = [];

member[0] = "Tamim";

member[1] = "Tahmid";

member[2] = 100;

document.getElementById("test").innerHTML =

member[1] + " " + member.length;

</script>

</body>

</html>

Copy

### ফলাফল

Tahmid 3

**সতর্কবাণী!!!**  
আপনি যদি ইনডেক্সিং-এ নাম ব্যবহার করেন জাভাস্ক্রিপ্ট অ্যারেকে স্ট্যান্ডার্ড অবজেক্ট হিসেবে পুনরায় ডিফাইন করে। তখন সকল অ্যারে মেথড এবং প্রোপার্টি ভুল ফলাফল দেখায়।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>আপনি অ্যারের ইন্ডেক্সিং নাম দিয়ে করলে জাভাস্ক্রিপ্ট অ্যারেকে একটি স্ট্যান্ডার্ড অবজেক্টে পরিণত করে এবং অ্যারের সকল প্রোপার্টি ও মেথড undefined বা ভুল ফলাফল রিটার্ন করে।</h3>

<p id="test"></p>

<script>

var member = [];

member["firstName"] = "Azizur";

member["lastName"] = "Rahman";

member["age"] = 32;

var a = member.length;

var b = member[0];

document.getElementById("test").innerHTML =

a + " " + b;

</script>

</body>

</html>

Copy

### ফলাফল

### আপনি অ্যারের ইন্ডেক্সিং নাম দিয়ে করলে জাভাস্ক্রিপ্ট অ্যারেকে একটি স্ট্যান্ডার্ড অবজেক্টে পরিণত করে এবং অ্যারের সকল প্রোপার্টি ও মেথড undefined বা ভুল ফলাফল রিটার্ন করে।

0 undefined

## অ্যারে এবং অবজেক্টের মধ্যে পার্থক্য

জাভাস্ক্রিপ্টে, অ্যারে নম্বর ইনডেক্স ব্যবহার করে।

জাভাস্ক্রিপ্টে, অবজেক্ট নাম ইনডেক্স ব্যবহার করে।

অ্যারে হচ্ছে একটি স্পেশাল অবজেক্ট যা নম্বর ইনডেক্স ব্যবহার করে।

## কখন অ্যারে আর কখন অবজেক্ট ব্যবহার করবেন।

* জাভাস্ক্রিপ্ট অ্যাসোসিয়েটিভ অ্যারে সাপোর্ট করে না।
* যখন আপনি এলিমেন্টের নাম স্ট্রিং(টেক্সট) আকারে চাইবেন তখন অবজেক্ট ব্যবহার করবেন।
* যখন আপনি এলিমেন্টের নাম নম্বর হিসেবে চাইবেন তখন আপনি অ্যারে ব্যবহার করবেন।

## new Array() পরিহার করা

এখানে জাভাস্ক্রিপ্টের বিল্ট-ইন অ্যারে কনস্ট্রাক্টর **new** Array() প্রয়োজন নেই।

**এর পরিবর্তে [] ব্যবহার করুন।**

এই দুইটি ভিন্ন স্টেটমেন্টই points নামের নতুন অ্যারে তৈরি করেছেঃ

var scores = new Array(30, 112, 50, 2, 8, 14); // খারাপ পদ্ধতি

var scores = [30, 112, 50, 2, 8, 14]; // ভালো পদ্ধতি

Copy

এই দুইটি ভিন্ন স্টেটমেন্টই ৬ সংখ্যার একটি নতুন অ্যারে তৈরি করেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> new Array() পরিবর্তে [] ব্যবহার করুন। </h3>

<p id="test"></p>

<script>

//var scores = new Array(30, 112, 50, 2, 8, 14); // খারাপ পদ্ধতি

var scores = [30, 112, 50, 2, 8, 14];

document.getElementById("test").innerHTML = scores[0];

</script>

</body>

</html>

Copy

### ফলাফল

### new Array() পরিবর্তে [] ব্যবহার করুন।

30

**new** কিওয়ার্ড কোডকে জটিল করে তুলে। এটি অপ্রত্যাশিত ফলাফল দেখাতে পারেঃ

var points = new Array(30, 112); // দুইটি এলিমেন্ট দিয়ে অ্যারে তৈরি (30 এবং 112)

Copy

যদি একটি এলিমেন্ট বাদ দেই?

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> new Array() ব্যবহার পরিহার করুন </h3>

<p id="test"></p>

<script>

var scores = new Array(30);

document.getElementById("test").innerHTML = scores[0];

</script>

</body>

</html>

Copy

### ফলাফল

### new Array() ব্যবহার পরিহার করুন

undefined

## একটি অ্যারেকে কিভাবে চিনা যায়

একটি সাধারণ প্রশ্ন হলোঃ একটি ভ্যারিয়েবল অ্যারে কিনা কিভাবে বুঝবো?

সমস্যা হচ্ছে জাভাস্ক্রিপ্টের **typeof** অপারেটর "object" রিটার্ন করেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>typeof অপারেটর, যখন একটি অ্যারে ব্যবহার করা হয়,তখন অবজেক্ট রিটার্ন করে। </h3>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = typeof animals;

</script>

</body>

</html>

Copy

### ফলাফল

### typeof অপারেটর, যখন একটি অ্যারে ব্যবহার করা হয়,তখন অবজেক্ট রিটার্ন করে।

object

typeof অপারেটর অবজেক্ট রিটার্ন করে কারন জাভাস্ক্রিপ্ট অ্যারে একটি অবজেক্ট।

### সমাধান ১

এই সমস্যা দূর করার জন্য ECMAScript 5 একটি নতুন মেথড **Array.isArray()** নির্ধারন করেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>এই সমস্যা দূর করার জন্য ECMAScript 5 একটি নতুন মেথড Array.isArray() নির্ধারন করেঃ </h3>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = Array.isArray(animals);

</script>

</body>

</html>

Copy

### ফলাফল

### এই সমস্যা দূর করার জন্য ECMAScript 5 একটি নতুন মেথড Array.isArray() নির্ধারন করেঃ

true

এই সমাধানের একটি সমস্যা রয়েছে, পুরাতন ব্রাউজারে ECMAScript 5 সাপোর্ট করে না।

### সমাধান ২

এই সমস্যা দূর করার জন্য আপনি একটি isArray() ফাংশন তৈরি করে নিতে পারেনঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>এই "হোম মেথড" ফাংশনটি যখন অ্যারেতে ব্যবহৃত হয়,তখন true রিটার্ন করে। </h3>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = isArray(animals);

function isArray(x) {

return x.constructor.toString().indexOf("Array") > -1;

}

</script>

</body>

</html>

Copy

### ফলাফল

### এই "হোম মেথড" ফাংশনটি যখন অ্যারেতে ব্যবহৃত হয়,তখন true রিটার্ন করে।

true

যদি আর্গুমেন্ট একটি অ্যারে হয় তাহলে উপরের ফাংশনটি সব সময় true রিটার্ন করবে।

### সমাধান ৩

যদি একটি অবজেক্ট কনস্ট্রাক্টরের মাধ্যমে তৈরি হয় তাহলে **instanceof** অপারেটর true রিটার্ন করেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>যখন একটি এরেতে instanceof অপারেটর ব্যবহৃত হয়, তখন true রিটার্ন করে। </h3>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals instanceof Array;

</script>

</body>

</html>

Copy

### ফলাফল

### যখন একটি এরেতে instanceof অপারেটর ব্যবহৃত হয়, তখন true রিটার্ন করে।

true

# জাভাস্ক্রিপ্ট অ্যারে মেথড

[« পূর্ববর্তী](https://www.sattacademy.com/js/js_arrays.php)

[পরবর্তী »](https://www.sattacademy.com/js/js_array_sort.php)

জাভাস্ক্রিপ্ট অ্যারের মূল শক্তি হলো অ্যারে মেথড।

## অ্যারেকে স্ট্রিং-এ রূপান্তর

জাভাস্ক্রিপ্ট toString () মেথডটি অ্যারে থেকে স্ট্রিং-এ রূপান্তর করে।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>toString() মেথডে একটি কমা দিয়ে স্ট্রিং-কে পৃথক করা হয়। </h3>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals.toString();

</script>

</body>

</html>

Copy

### ফলাফল

join() মেথড সকল অ্যারে এলিমেন্টকে একটি স্ট্রিং-এ যুক্ত করে।

এটা toString() এর মত কাজ করেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> join() মেথড একটি স্ট্রিং-কে অ্যারে এলিমেন্টে সংযুক্ত করে। </h3>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals.join(" \* ");

</script>

</body>

</html>

Copy

### ফলাফল

## pop() এবং push()

অ্যারে নিয়ে কাজ করার সময় খুব সহজেই এলিমেন্ট বাদ দিতে এবং নতুন এলিমেন্ট যোগ করতে পারবেন।

pop() মেথড দিয়ে অ্যারে থেকে এলিমেন্ট বাদ দেয়া হয় এবং push() মেথড দিয়ে অ্যারেতে এলিমেন্ট যোগ করা হয়।

## pop() মেথড

pop() মেথড দিয়ে অ্যারের শেষ এলিমেন্ট বাদ দেয়া হয়ঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> pop মেথড একটি অ্যারে থেকে শেষ এলিমেন্টকে সরায়। </h3>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals;

function myFunc() {

animals.pop();

document.getElementById("test").innerHTML = animals;

}

</script>

</body>

</html>

Copy

### ফলাফল

pop() মেথড বাদ দেওয়া ভ্যালুকে রিটার্ন করেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> pop মেথড সরানো আইটেমকে রিটার্ন করে। </h3>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals;

function myFunc() {

document.getElementById("test").innerHTML = animals.pop();

}

</script>

</body>

</html>

Copy

### ফলাফল

## Push()

push() মেথড দ্বারা একটি অ্যারের শেষে নতুন এলিমেন্ট যুক্ত করা হয়ঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> push মেথড একটি অ্যারে নতুন এলিমেন্টকে সংযুক্ত করে। </h3>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals;

function myFunc() {

animals.push("Elephant");

document.getElementById("test").innerHTML = animals;

}

</script>

</body>

</html>

Copy

### ফলাফল

push() মেথড অ্যারে এর নতুন দৈর্ঘ্য রিটার্ন করেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> push মেথড নতুন array লেন্থকে রিটার্ন করে। </h3>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals;

function myFunc() {

document.getElementById("test").innerHTML = animals.push("Elephant");

}

</script>

</body>

</html>

Copy

### ফলাফল

## shift() মেথড এবং unshift() মেথড

shift() মেথড এবং pop() মেথড একই রকম কাজ করে, শুধুমাত্র শেষের এলিমেন্টের পরিবর্তে প্রথম এলিমেন্টকে বাদ দেয়ঃ

**shift()** মেথড অ্যারের প্রথম এলিমেন্ট বাদ দেয় এবং সকল এলিমেন্টের ইনডেক্স কমিয়ে দেয়ঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>shift মেথড একটি অ্যারের প্রথম এলিমেন্ট এবং অন্য সকল এলিমেন্টকে একটি একটি করে মুছে ফেলে </h3>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals;

function myFunc() {

animals.shift();

document.getElementById("test").innerHTML = animals;

}

</script>

</body>

</html>

Copy

### ফলাফল

unshift() মেথড অ্যারের শুরুতে একটি নতুন এলিমেন্ট যুক্ত করেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> unshift মেথড একটি অ্যারের শুরুতে নতুন এলিমেন্টকে সংযুক্ত করে। </h3>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals;

function myFunc() {

animals.unshift("Elephant");

document.getElementById("test").innerHTML = animals;

}

</script>

</body>

</html>

Copy

### ফলাফল

unshift() মেথড অ্যারের নতুন দৈর্ঘ্য রিটার্ন করে।

## এলিমেন্ট পরিবর্তন

অ্যারে এলিমেন্ট ইনডেক্স নম্বর ব্যবহার করে এক্সেস করা হয়ঃ

অ্যারে ইনডেক্স ০ দিয়ে শুরু হয়, [০] হচ্ছে অ্যারের প্রথম এলিমেন্ট, [১] হচ্ছে দ্বিতীয়, [২] হচ্ছে তৃতীয়...

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>অ্যারে এলিমেন্ট তাদের ইনডেক্স নাম্বার অনুযায়ী এক্সেস হয়। </h3>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals;

function myFunc() {

animals[0] = "Elephant";

document.getElementById("test").innerHTML = animals;

}

</script>

</body>

</html>

Copy

### ফলাফল

অ্যারেতে নতুন এলিমেন্ট যোগ করার আরেকটি সহজ পদ্ধতি হচ্ছে length প্রোপার্টি ব্যবহার করাঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> length প্রোপার্টি push() মেথড ছাড়াই একটি অ্যাারেতে নতুন এলিমেন্ট সহজ পদ্ধতিতে সংযুক্ত করে। </h3>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals;

function myFunc() {

animals[animals.length] = "Elephant";

document.getElementById("test").innerHTML = animals;

}

</script>

</body>

</html>

Copy

### ফলাফল

## এলিমেন্ট ডিলেট করা

যেহেতু জাভাস্ক্রিপ্ট অ্যারে একটি অবজেক্ট, তাই জাভাস্ক্রিপ্ট **delete** অপারেটর ব্যবহার করে অ্যারের এলিমেন্ট ডিলেট করা যায়ঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> মুছে ফেলা এলিমেন্টগুলো একটি অ্যারেতে ছেড়ে দিলে undefined হয়। </h3>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals;

function myFunc() {

delete animals[0];

document.getElementById("test").innerHTML = animals[0];

}

</script>

</body>

</html>

Copy

### ফলাফল

## splice() মেথড

splice() মেথড একটি অ্যারেতে নতুন এলিমেন্ট যোগ করার জন্য ব্যবহার করা হয়ঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> splice() মেথড একটি এরেতে নতুন এলিমেন্টকে সংযুক্ত করে। </h3>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals;

function myFunc() {

animals.splice(3, 0, "Elephant", "Goat");

document.getElementById("test").innerHTML = animals;

}

</script>

</body>

</html>

Copy

### ফলাফল

প্রথম প্যারামিটারে এলিমেন্টের ইনডেক্স নম্বর নির্ধারন করা হয় যেখানে এলিমেন্টকে রাখা হবে।

দ্বিতীয় প্যারামিটারে কতটি এলিমেন্ট বাদ দেওয়া হবে তা নির্দেশ করা হয়।

## splice() ব্যবহার করে এলিমেন্ট বাদ দেওয়া

বুদ্ধিমত্তার সাথে প্যারামিটার সেটিং-এর মাধ্যমে আপনি splice() মেথড ব্যবহার করে অ্যারের "undefined" এঁরর ছাড়াই এলিমেন্ট বাদ দিতে পারবেনঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> splice() মেথড এরে এলিমেন্টকে সরাতে ব্যবহৃত হয়। </h3>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals;

function myFunc() {

animals.splice(0, 1);

document.getElementById("test").innerHTML = animals;

}

</script>

</body>

</html>

Copy

### ফলাফল

প্রথম প্যারামিটারে এলিমেন্টের ইনডেক্স নম্বর নির্ধারন করা হয় যেখানে এলিমেন্টকে রাখা হবে।

দ্বিতীয় প্যারামিটারে কতটি এলিমেন্ট বাদ দেওয়া হবে তা নির্দেশ করা হয়।

বাকি প্যারামিটার বাদ দেওয়া হয়েছে। তার মানে নতুন কোন এলিমেন্ট যোগ করা হবে না।

## অ্যারে যুক্ত করা

concat() মেথড দুইটি অ্যারেকে যুক্ত করে একটি নতুন অ্যারে তৈরি করেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h2>দুইটি অ্যারেকে সংযুক্ত করা</h2>

<p id="test"></p>

<script>

var girls = ["Safa", "Rimu"];

var boys = ["Sarif", "Motiur","Imran"];

var children = girls.concat(boys);

document.getElementById("test").innerHTML = children;

</script>

</body>

</html>

Copy

### ফলাফল

concat() মেথড আর্গুমেন্ট হিসেবে একাধিক অ্যারেও নিতে পারেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h2>তিনটি অ্যারেকে সংযুক্ত করবে </h2>

<p id="test"></p>

<script>

var arr1 = ["Safa", "Rimu"];

var arr2 = ["Sarif", "Motiur","Imran"];

var arr3 = ["Tamim", "Tahmid"];

var children = arr1.concat(arr2, arr3);

document.getElementById("test").innerHTML = children;

</script>

</body>

</html>

Copy

### ফলাফল

## অ্যারে আলাদা(slice) করা

**slice()** মেথড একটি অ্যারের অংশকে আলাদা করে নতুন একটি অ্যারে তৈরি করে।

এই উদাহরণে একটি অ্যারের প্রথম এলিমেন্ট("Lion") থেকে আলাদা করে।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>এই slice() মেথড উদাহরণে একটি অ্যারের প্রথম এলিমেন্ট (জাম) থেকে আলাদা করা শুরু করে। </h3>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

var sli = animals.slice(1);

document.getElementById("test").innerHTML = animals + "<br>" + sli;

</script>

</body>

</html>

Copy

### ফলাফল

slice() মেথড একটি নতুন অ্যারে তৈরী করে। এটি সোর্স অ্যারে থেকে কোনো এলিমেন্ট বাদ দেয় না।

slice() মেথড দুইটি আর্গুমেন্টও নিতে পারেঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>slice() মেথড slice(1,3) এর মত দুইটি আরগুমেন্ট নিতে পারেঃ</h3>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

var sli = animals.slice(1,2);

document.getElementById("test").innerHTML = animals + "<br>" + sli;

</script>

</body>

</html>

Copy

### ফলাফল

যদি শেষ আর্গুমেন্ট না দেওয়া হয় তাহলে slice() মেথড অ্যারের বাকি এলিমেন্টকে আলাদা করে নেয়।

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>এই slice() উদাহরণস্বরূপ একটি অ্যারেকে অ্যারের এলিমেন্ট 2 ( "Tiger") থেকে শুরু করে একটি অংশ আলাদা করবেঃ

</h3>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

var sli = animals.slice(2);

document.getElementById("test").innerHTML = animals + "<br>" + sli;

</script>

</body>

</html>

Copy

### ফলাফল

## valueOf() মেথড

valueOf() মেথড অ্যারের একটি ডিফল্ট মেথড। এটি একটি অ্যারেকে প্রিমিটিভ ভ্যালুতে রুপান্তর করে।

যখন একটি প্রিমিটিভ ভ্যালুর দরকার হয় জাভাস্ক্রিপ্ট স্বয়ংক্রিয়ভাবে একটি অ্যারেকে একটি স্ট্রিং-এ রূপান্তর করে।

এই কারণে, নিচের সকল উদাহরণ একই ফলাফল দেখায়ঃ

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>জাভাস্ক্রিপ্ট স্বয়ংক্রিয়ভাবে একটি প্রত্যাশিত পুরনো ভ্যালুকে অ্যারে থেকে স্ট্রিং-এ পরিবর্তন করে। </h3>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals;

</script>

</body>

</html>

Copy

### ফলাফল

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3> valueOf() একটি অবজেক্টে পুরনো ভ্যালুকে রিটার্ন করে। </h3>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals.valueOf();

</script>

</body>

</html>

Copy

### ফলাফল

### উদাহরণ

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h3>toString() মেথডে একটি কমা দিয়ে স্ট্রিং-কে পৃথক করা হয়। </h3>

<p id="test"></p>

<script>

var animals = ["Lion", "Tiger", "Horse", "Hen"];

document.getElementById("test").innerHTML = animals.toString();

</script>

</body>

</html>

Copy

### ফলাফল

সকল জাভাস্ক্রিপ্ট অবজেক্টের একটি valueOf() এবং একটি toString() মেথড আছে।

Bottom of Form