

This app works best with JavaScript enabled.



হাতেকলমে জাভাস্ক্রিপ্ট



জাভাস্ক্রিপ্টঃ ব্যাসিক অবজেক্ট

আমরা গত পর্বে অ্যাারে নিয়ে আলোচনা করেছি। অ্যাারেতে আইটেমগুলো ইনডেক্স নাম্বার দিয়ে অ্যাক্সেস করতে হয়। এবং সেজন্যে অ্যাারেতে থাকা আইটেমগুলোর সিরিয়াল ঠিক রাখা জরুরী। কিন্তু আমরা যদি ইনডেক্স ছাড়া স্পেসিফিক কোনো কীওয়ার্ড দিয়ে এরকমভাবে অ্যারেরমতো কোনো ডাটা স্ট্রাকচার থেকে আইটেম অ্যাক্সেস করতে চাই তাহলে কি উপায় থাকতে পারে? হ্যাঁ একটা উপায় আছে, সেটা হচ্ছে অবজেক্ট। অবজেক্ট ইউজ করা অনেকটা হ্যান্ডি এবং অ্যাারে থেকেও আরো অনেক ফ্লেক্সিবল।

ধরুন আপনি একজনের কিছু ইনফরমেশন স্টোর করতে চাচ্ছেন। এখন ইনফরমেশনগুলো কোনো না কোনোভাবে একে অপরের সাথে সম্পর্কযুক্ত। সেক্ষেত্রে আমরা পৃথক পৃথক ভ্যারিয়েবল না নিয়ে অ্যাারে

ইউজ করতে পারি। কিন্তু তারপরেও অ্যারেতে একটা সমস্যা, অ্যারেতে আমরা আইটেম অ্যাক্সেস করতে চাইলে ইনডেক্স নাম্বার দিয়ে করতে হয় যেটা খুব অর্থপূর্ণ না। যদি এমন কিছু হয় তাহলে কেমন হয়ঃ

```
var rahim = {  
    পুরোনাম: "রহিম",  
    বয়স: 30,  
    পেশা: "ডাক্তার",  
    ঠিকানা: "ঢাকা",  
    ফোন: "১২৩৪৫৬৭৮৯০"  
}
```

এখানে ধরুন রহিম একটা অবজেক্ট, তাহলে তার পুরোনাম, বয়স, ঠিকানা, পেশা এগুলো তার প্রপার্টিজ। এইখানে এগুলো একটা আরেকটা সাথে সম্পর্কযুক্ত। এক্ষেত্রে এরকম ডাটা স্ট্রাকচার পারফেক্ট কাজ করে। আর এখন রহিমের এই ইনফরমেশন বা প্রপার্টিজ গুলো অ্যাক্সেস করতে চাইলে, রহিমের পুরোনাম চাইলে `rahim.purona` বা রহিমের বয়স চাইলে `rahim.bay` লিখলেই চলে আসবে। হ্যা জাভাস্ক্রিপ্ট এর অবজেক্ট সেইমভাবেই কাজ করে। এভাবে অ্যারের মতো ইনডেক্স নাম্বার ইউজ না করে কীওয়ার্ড ইউজ করে আপনি আইটেমগুলো অ্যাক্সেস করতে পারবেন।

এভাবে একটা অবজেক্ট ডিক্লেয়ার করা যায় কয়েকভাবে। যেমন এভাবেও করা যায় যেটা একটু কঠিন আর তত বেশী ইউজও করা হয় নাঃ

```
var rahim = new Object()
```

তবে এভাবে করা সবচেয়ে ইজি এবং এভাবেই সাধারণত বেশী ইউজ করা হয়ঃ

```
var rahim = {  
  fullName: 'Rahim Miya',  
  age: 21,  
  address: 'Dhaka',  
  job: 'Job Holder'  
}
```

এবার পুরো অবজেক্ট অ্যাক্সেস করতে চাইলে জাস্ট অবজেক্ট এর নামটা লিখুনঃ

```
rahim
```

আউটপুট দেখুনঃ

```
> var rahim = {  
  fullName: 'Rahim Miya',  
  age: 21,  
  address: 'Dhaka',  
  job: 'Job Holder'  
}  
< undefined  
> rahim  
< ► {fullName: "Rahim Miya", age: 21, address: "Dhaka", job: "Job Holder"}  
> |
```

বিঃদ্রঃ ক্রোমের কন্সোলে মাল্টিলাইন কোড লিখতে গেলে দেখবেন সেটা এন্টার হয়ে এক্সিকিউট হয়ে যায়। তবে আপনি এটা স্কিপ করে মাল্টিলাইন কোড লিখতে পারবেন জাস্ট **shift + Enter** একসাথে চেপে। তাহলে কোড এক্সিকিউট না হয়ে পরের লাইনে যাবে যেখানে আপনি আপনার কোড

লিখতে পারবেন।

এবার দেখুন এখানেও পুরো অবজেক্ট দেখাচ্ছে। বাম পাশের ত্রিভুজাকৃতির বাটনে ক্লিক করলে এখানেও সব দেখতে পারবেন। কিন্তু ইনডেক্স নাম্বারের পরিবর্তে কীওয়ার্ড দেখতে পারবেন যেগুলো ইউজ করে আপনি এই অবজেক্ট এর আইটেমগুলো অ্যাক্সেস করতে পারবেনঃ

```
> rahim
< ▼ {fullName: "Rahim Miya", age: 21, address: "Dhaka", job: "Job Holder"} ⓘ
  address: "Dhaka"
  age: 21
  fullName: "Rahim Miya"
  job: "Job Holder"
  ► __proto__: Object
>
```

এবার আসি কিভাবে অবজেক্ট অ্যাক্সেস করবেন। আমি গত পর্বে যদিও এটা নিয়ে আলোচনা করেছি, তবে এখানে এটা বেশী কাজে আসবে। আমরা দুইভাবে একটা অবজেক্ট থেকে আইটেম অ্যাক্সেস করতে পারি

১। ডট নোটেশান

২। ব্র্যাকেটস নোটেশান

এখন চলুন এই নোটেশানের বিস্তারিত আলোচনায় যাইঃ

১। ডট নোটেশানঃ এখানে ডট দিয়ে একটা অবজেক্ট এর ভিতরের আইটেম অ্যাক্সেস করা হয়। এবং জাভাস্ক্রিপ্ট এ সাধারণত এই নোটেশানটাই বেশী ইউজ করা হয়। **rahim** এর পুরোনাম ডট নোটেশানে অ্যাক্সেস করতে চাইলে অবজেক্ট এর নাম তারপর ডট দিয়ে পুরোনামের কীওয়ার্ড দিলেই হবেঃ

```
rahim.fullName
```

```
> rahim
< ▶ {fullName: "Rahim Miya", age: 21, address: "Dhaka", job: "Job Holder"}
> rahim.fullName
< "Rahim Miya"
> |
```

২। ব্র্যাকেটস নোটেশানঃ আবার ব্র্যাকেটস নোটেশানেও অবজেক্ট এর আইটেম অ্যাক্সেস করতে পারবেনঃ

```
rahim['fullName']
```

```
> rahim
< ▶ {fullName: "Rahim Miya", age: 21, address: "Dhaka", job: "Job Holder"}
> rahim['fullName']
< "Rahim Miya"
> |
```

এখন একটা অবজেক্ট ডিক্লেয়ার করা হয়ে গেছে। এখন যদি এখানে নতুন কোনো আইটেম অ্যাড করতে চাই তাহলে কি উপায়? হ্যাঁ আমরা দেখেছি যে অবজেক্ট এর কোন ইনডেক্স নাম্বার নাই। তাই এখানে একটার পর একটা আইটেমের সিরিয়াল মেইন্টেইন করা জরুরী না। আমরা জাস্ট কীওয়ার্ড দিয়েই যেকোনো আইটেম অ্যাক্সেস করতে পারি। একবার অবজেক্ট ডিক্লেয়ার করা হয়ে গেলে পড়ে যদি আমরা নতুন কোনো আইটেম অ্যাড করতে চাইঃ

```
rahim.zipCode = 3517
```

```
> rahim
< ► {fullName: "Rahim Miya", age: 21, address: "Dhaka", job: "Job Holder"}
> rahim.zipCode = 3517
< 3517
> rahim
< ► {fullName: "Rahim Miya", age: 21, address: "Dhaka", job: "Job Holder", zipCode: 3517}
> rahim.zipCode
< 3517
> |
```

কি সুন্দর করে নতুন আইটেম অ্যাড হয়ে গেলো। এখন আমরা যদি অলরেডি আছে এমন একটা আইটেমের ভ্যালু চ্যাঞ্জ করতে চাই:

```
rahim.job = 'Student'
```

```
> rahim
< ► {fullName: "Rahim Miya", age: 21, address: "Dhaka", job: "Job Holder", zipCode: 3517}
> rahim.job = 'Student'
< "Student"
> rahim
< ► {fullName: "Rahim Miya", age: 21, address: "Dhaka", job: "Student", zipCode: 3517}
> rahim.job
< "Student"
>
```

ব্যাস এভাবেই সবকিছু অনেক সহজ জাভাস্ক্রিপ্ট এর অবজেক্ট এ। অ্যারে থেকে অনেক ফ্লেক্সিবল।

এবার আসি অবজেক্ট এর স্পেশালিটি নিয়ে। আপনি কন্সোলে অবজেক্ট খুলে `proto` এর ভিতরে অনেক ফাংশন পাবেন যেগুলো আপনি এই অবজেক্ট এর সাথে ইউজ করতে পারবেনঃ

```
> rahim
< {fullName: "Rahim Miya", age: 21, address: "Dhaka", j
  address: "Dhaka"
  age: 21
  fullName: "Rahim Miya"
  job: "Student"
  zipCode: 3517
  __proto__:
    ▶ constructor: f Object()
    ▶ hasOwnProperty: f hasOwnProperty()
    ▶ isPrototypeOf: f isPrototypeOf()
    ▶ propertyIsEnumerable: f propertyIsEnumerable()
    ▶ toLocaleString: f toLocaleString()
    ▶ toString: f toString()
    ▶ valueOf: f valueOf()
    ▶ __defineGetter__: f __defineGetter__()
    ▶ __defineSetter__: f __defineSetter__()
    ▶ __lookupGetter__: f __lookupGetter__()
    ▶ __lookupSetter__: f __lookupSetter__()
    ▶ get __proto__: f __proto__()
    ▶ set __proto__: f __proto__()
```

তবে `proto` এর ভিতরের সবকিছু এখনি বুঝতে পারবেন না হয়তো, কিন্তু জাস্ট মাথায় রাখেন যে এভাবে খুলে খুলে সব দেখতে পারবেন কন্সোলে। যেটা অনেক কাজে লাগে কোনো কিছু ডেভেলপমেন্ট এর সময়।

এখন অবজেক্ট এ আরো স্পেশাল যে ব্যাপারটা সেটা হচ্ছে আপনি চাইলে এখানে ফাংশন ও সেইভ করে রেখে দিতে পারবেন আইটেম হিসেবে। ফাংশন কি বুঝতে না পারলে আমি পরের পর্বে এটা নিয়ে ডিসকাস করবো আপাতত না বুঝলেও মাথায় রাখবেন যে অবজেক্ট এর ভিতরে ফাংশনকে আইটেম হিসেবে নেওয়া যায়ঃ

```
rahim.welcomeMsg = function() {
```



```
    console.log('Hello There');  
  }
```

```
> rahim  
< ▶ {fullName: "Rahim Miya", age: 21, address: "Dhaka", job: "Student", zipCode: 3517}  
> rahim.welcomeMsg = function() {  
    console.log('Hello There');  
  }  
< f () {  
    console.log('Hello There');  
  }  
> rahim  
< ▼ {fullName: "Rahim Miya", age: 21, address: "Dhaka", job: "Student", zipCode: 3517, ...} ⓘ  
    address: "Dhaka"  
    age: 21  
    fullName: "Rahim Miya"  
    job: "Student"  
    ▶ welcomeMsg: f ()  
    zipCode: 3517  
    ▶ __proto__: Object  
>
```

এখন সেই ফাংশন কে অ্যাক্সেস করতে চাইলে সেইমভাবেই বাট একটু অন্যরকম ভাবে করতে হবে।

কারণ ফাংশন কল করতে হলে ফাংশ্ট ব্র্যাকেট ইউজ করতে হয়ঃ

```
rahim.welcomeMsg()
```

```

> rahim
< ▼ {fullName: "Rahim Miya", age: 21, address: "Dhaka", job: "Student", zipCode: 3517, ...} ⓘ
  address: "Dhaka"
  age: 21
  fullName: "Rahim Miya"
  job: "Student"
  ▶ welcomeMsg: f ()
  zipCode: 3517
  ▶ __proto__: Object
> rahim.welcomeMsg()
Hello There
< undefined
>

```

এখন উপরের এই ফাংশনটা অবজেক্ট ডিফাইন করার সময় ভ্যারিয়েবলের মতো করেও ফাংশন স্টেটমেন্ট আকারে নিতে পারবো আমরা এভাবেঃ

```

var rahim = {
  fullName: 'Rahim Miya',
  age: 21,
  address: 'Dhaka',
  job: 'Job Holder',
  welcomeMsg: function() {
    console.log('Hello There');
  }
}

```

দুইটাই এক। প্রথমটায় আমরা অবজেক্ট ডিক্লেয়ার করে ফেলেছি, পরে ফাংশন অ্যাসাইন করেছি সেটোতে। আর দ্বিতীয়টায় অবজেক্ট ডিফাইন করার সময়েই ফাংশনটাও দিয়ে দিয়েছি অন্যান্যগুলোর মতোই।

অবজেক্ট এর সবগুলো আইটেম অ্যাক্সেস করতে চাইলে? হ্যাঁ তাইলে মনে হয় লুপ ইউজ করতে হবে। কিন্তু সমস্যা হলো এখানে তো ইনডেক্স নাম্বার নাই। লুপ দিয়ে আইটেম অ্যাক্সেস করতে ত লুপ

লাগবে...? কিন্তু না। আরেকধরনের স্পেশাল লুপ আছে যেটা ইউজ করে আপনি সবগুলো আইটেম অ্যাক্সেস করতে পারবেন। যদিও এটা তত কাজের না তাই আমি লুপের পর্বে বিশেষ করে আলোচনা করিনি।

```
for(item in rahim) {  
    console.log(item)  
}
```

```
> rahim  
< ▶ {fullName: "Rahim Miya", age: 21, address: "Dhaka", job: "Student", zipCode: 3517, ...}  
> for(item in rahim) {  
    console.log(item)  
}  
fullName  
age  
address  
job  
zipCode  
welcomeMsg  
< undefined  
> |
```

এখন অবজেক্ট এতোই ফ্লেক্সিবল যে আপনি চাইলে অবজেক্ট এর ভিতরে অ্যারেও ইউজ করতে পারবেন আবার অ্যারের ভিতরেও অবজেক্ট ইউজ করতে পারবেন।

অবজেক্ট এর ভিতরে অ্যারেঃ

```
var objArr = {  
    normal: 'Normal Item',  
    name: ['Rahim', 'Karim', 'Rafiq', 'Shafiq'],  
}
```

```
    age: [10, 20, 30, 40, 50]
  }
```

```
> objArr
< ▼ {normal: "Normal Item", name: Array(4), age: Array(5)} ⓘ
  ▶ age: (5) [10, 20, 30, 40, 50]
  ▶ name: (4) ["Rahim", "Karim", "Rafiq", "Shafiq"]
    normal: "Normal Item"
  ▶ __proto__: Object
```

ভিতরের আইটেমগুলোও ঠিকঠাক মতো অ্যারে আর অবজেক্ট এর নিয়ম-কানুন মেনেই অ্যাক্সেস করতে পারবেন। ধরুন আমি রাফিকের নাম চাচ্ছিঃ

```
objArr.name[2]
```

```
> objArr
< ▼ {normal: "Normal Item", name: Array(4), age: Array(5)} ⓘ
  ▶ age: (5) [10, 20, 30, 40, 50]
  ▶ name: (4) ["Rahim", "Karim", "Rafiq", "Shafiq"]
    normal: "Normal Item"
  ▶ __proto__: Object

> objArr.name[2]
< "Rafiq"
```

অ্যারের ভিতরে অবজেক্টঃ

আমরা জানি অ্যারেতে আমরা অনেকগুলো আইটেম স্টোর করতে পারি। এখন সে আইটেম আরেকটা অ্যারেও হতে পারে, নাম্বার, স্ট্রিং, বুলিয়ান, ফাংশন বা আরেকটা অবজেক্ট ও হতে পারেঃ

```
var arrObj = ['Zonayed', {name: 'Rafiq', age: 21, job: 'Student'}, 'Bangladesh']
```

```
> var arrObj = ['Zonayed', {name: 'Rafiq', age: 21, job: 'Student'}, 'Bangladesh']
< undefined
> arrObj
< ▼ (3) ["Zonayed", {...}, "Bangladesh"] ⓘ
  0: "Zonayed"
  ▶ 1: {name: "Rafiq", age: 21, job: "Student"}
  2: "Bangladesh"
  length: 3
  ▶ __proto__: Array(0)
> |
```

এখানে অবজেক্ট এর ইনডেক্স নাম্বার কতো বলুন তো? হ্যা অবজেক্ট এর ইন্ডেক্স নাম্বার ১ এবং তারপরে বাংলাদেশের ইনডেক্স নাম্বার ২ এ। আর এই অবজেক্ট থেকে আইটেম অ্যাক্সেস করতে চাইলেঃ

```
arrObj[1].name
```

```
> var arrObj = ['Zonayed', {name: 'Rafiq', age: 21, job: 'Student'}, 'Bangladesh']
< undefined
> arrObj
< ▼ (3) ["Zonayed", {...}, "Bangladesh"] ⓘ
  0: "Zonayed"
  1: {name: "Rafiq", age: 21, job: "Student"}
  2: "Bangladesh"
  length: 3
  __proto__: Array(0)
> arrObj[1].name
< "Rafiq"
>
```

এভাবে আপনি অ্যারের ভিতরে অবজেক্ট, আবার সেই অবজেক্ট ভিতরে অ্যারে, আবার সেই অ্যারের ভিতরে অবজেক্ট বা অবজেক্ট এর ভিতরে অ্যারে, আবার সেই অ্যারের ভিতরে অবজেক্ট, আবার সেই অবজেক্টের ভিতরে অ্যারে নিতে পারবেন। আর আপনি যতক্ষণ পর্যন্ত অ্যারের বেলায় অ্যারের নিয়ম-কানুন আর অবজেক্ট এর বেলায় অবজেক্ট এর নিয়ম-কানুন মানছেন ততক্ষণ আপনি এভাবে নেস্টেড করে ডাটা স্ট্রাকচার বানাতেও পারবেন আবার ভিতর থেকে আইটেম অ্যাক্সেসও করতে পারবেন। অনেকটাই ফ্লেক্সিবল তাই না...? হ্যা তাই [?]

আজকে এই পর্যন্তই। ভালো থাকবেন।

আপনার মন্তব্যঃ

যদি এই পোস্টে কোন ভুল(যেকোনো ধরনের) পেয়ে থাকেন অথবা কোনো ব্যাপারে সন্দেহ থাকে তাহলে এখানে জানাতে পারবেন।



জাভাস্ক্রিপ্ট ব্যাসিক



জাভাস্ক্রিপ্ট অ্যাডভান্স



জাভাস্ক্রিপ্ট ইএস৬



জাভাস্ক্রিপ্ট ডম ম্যানিপুলেশন



নিত্যদিনের জাভাস্ক্রিপ্ট

জাভাস্ক্রিপ্ট অ্যালগরিদম ও ডাটা স্ট্রাকচার



জাভাস্ক্রিপ্ট সফট স্কিল

সম্পর্কেঃ

প্রোজেক্টটি সম্পূর্ণ সোর্স কোডসহ গিটহাবে রয়েছে। আপনার ভালো লেগে থাকলে স্টার দিয়ে আসবেন। আপনার পরামর্শ, মন্তব্য এবং ভুলত্রুটি গিটহাবে ইস্যু করে দেওয়ার জন্যে অনুরোধ থাকলো

আপনার জন্যঃ

রিঅ্যাক্ট জেএস শিখুন

আমি মিডিয়ামে

আমার ব্লগ

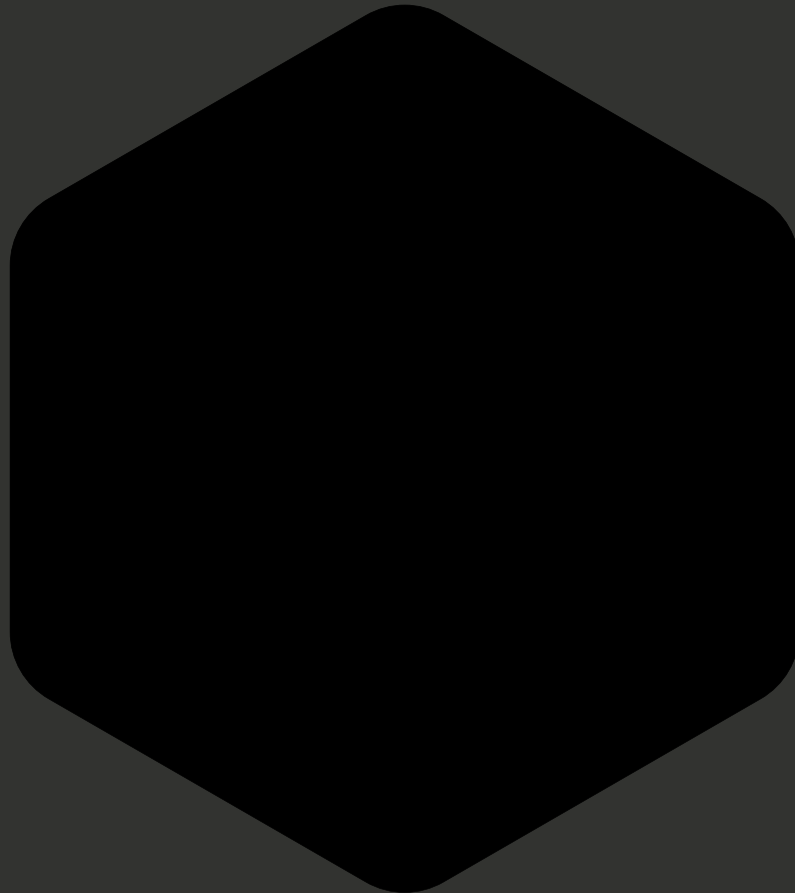
রিসোর্সঃ

এমডিএন ডকুমেন্টেশন

ইউডেমি কোর্স

Eloquent JavaScript

You Don't Know JS



♥ এর সাথে ডেভেলপ করেছে **জুনায়েদ আহমেদ**