

This app works best with JavaScript enabled.



হাতেকলমে জাভাস্ক্রিপ্ট



জাভাস্ক্রিপ্টঃ অ্যারে নিয়ে সবকিছু

আপনি ধরুন অনেকগুলো ডাটা/ভ্যারিয়েবল স্টোর করতে চাচ্ছেন। এজন্যে আপনি কি করতে পারেনঃ

```
var name1 = 'Zonayed Ahmed';  
var name2 = 'Zawad Ahmed';  
var name3 = 'Zobayer Ahmed';  
var name4 = 'Masood Ahmed';
```

```
var name5 = 'Ahmed Zonayed';
```

এখন দেখুন এখানে অনেকগুলো ভ্যারিয়েবল স্টোর করতে চাচ্ছি, আর তারজন্যে প্রত্যেকবারই একটা একটা করে নাম নিয়ে নিয়ে ভ্যারিয়েবলগুলো স্টোর করতে হচ্ছে। আবার এখানে ডাটা সবগুলো একই টাইপের অনেকটা। কিছু নাম আর কি। এখন এভাবে অবশ্যই খুব বেশী ভালো দেখাচ্ছে না। হয়তো দুই একটা ভ্যারিয়েবলের জন্যে এভাবে আলাদা আলাদা করে ভ্যারিয়েবল নিয়ে ডাটা স্টোর করা ঠিক আছে। কিন্তু যখন অনেকগুলো হয়ে যায় তখন এটার চাইতেও ভালো সমাধান আছে। যেটা হচ্ছে অ্যারে। আমি আজকে অ্যারে নিয়েই আলোচনা করবো।

ধরুনঃ

```
□□□ = □□□□, □□□□, □□□□□, □□□□, □□□□□□□, □□□□
```

শফিকের নাম চাইঃ

```
□□□[□ □□□□□□□□□ □□]
```

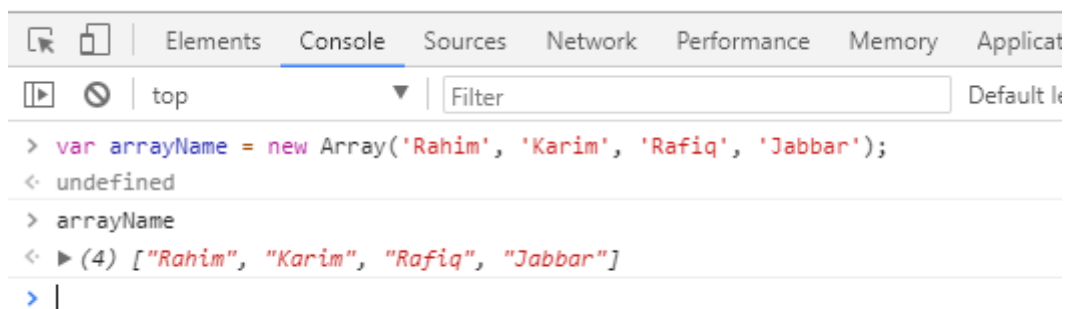
এভাবেই কাজ করে অ্যারে অনেকটা। যদিও আসল কোড থেকে উদাহরন দিলে ক্লিয়ার হয়ে যাবে। তবে আসল কনসেপ্ট ক্লিয়ার হওয়া জরুরী। অ্যারে নন-প্রিমিটিভ/রেফারেন্স ডাটা টাইপ। আমি এগুলো নিয়ে পূর্বে লিখেছিলাম। অ্যারে একধরনের অবজেক্ট। আর অবজেক্ট মানে কি জানি? হ্যা, অবজেক্ট মানে এটার কিছু প্রপার্টিজ থাকবে। আপনি অবজেক্ট হলে আপনার হাত, কান, গলা, চোখ ইত্যাদি ইত্যাদি আপনার প্রপার্টিজ। সেভাবে অ্যারেরও কিছু প্রপার্টিজ আছে যেগুলো ইউজ করে আপনি অ্যারে থেকে অনেক কিছু

অ্যাক্সেস করতে পারবেন। অ্যারে সম্পর্কে জানতে পারবেন। অ্যারে কে মডিফাই করতে পারবেন। আমি এগুলো সবকিছু নিয়েই আজকে আলোচনা করবো

অ্যারে ডিক্লেয়ার করাঃ অ্যারে দুইভাবে ডিক্লেয়ার করা যায়। তার মধ্যে প্রথমটা খুব বেশী ইউজ করা হয় না কিন্তু জেনে রাখা ভালো

```
var arrayName = new Array('Rahim', 'Karim', 'Rafiq', 'Jabbar');
```

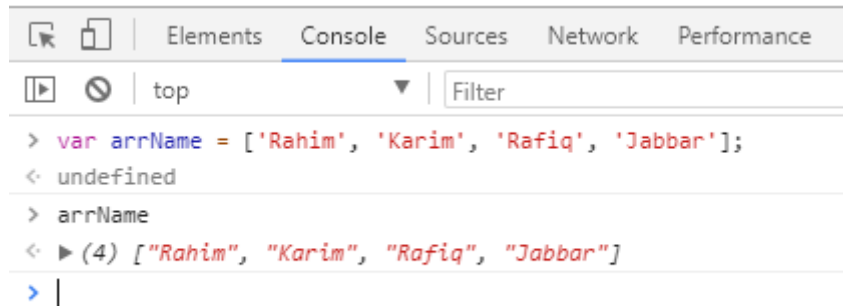
এটার আউটপুট আসবেঃ



ক্রোম কন্সোলেই সরাসরি ওপেন করে লিখে প্র্যাক্টিস করতে পারবেন। দেখুন এখানে আউটপুট এসেছে আপনার ভিতরে দেওয়া সবগুলো ডাটা। এখন এইভাবে ডিক্লেয়ার করাটা খুব বেশী পপুলার না। আর একটু কঠিনও বটে। আপনি নরমালিও করতে পারবেনঃ

```
var arrName = ['Rahim', 'Karim', 'Rafiq', 'Jabbar'];
```

এভাবেও সহজেই করে ফেলা যায়। এবার এটাও ক্রোম কন্সোলে লিখে আউটপুট দেখুনঃ



```
> var arrName = ['Rahim', 'Karim', 'Rafiq', 'Jabbar'];  
< undefined  
> arrName  
< ▶ (4) ["Rahim", "Karim", "Rafiq", "Jabbar"]  
> |
```

Same Output

এ তো গেলো ডিক্লেয়ারেশনের পর্ব। এবার আসি আরো কিছু গভীর জিনিস নিয়ে। আগে বলেছিলাম ক্রোম ডেভেলপার কন্সোল আপনাকে অনেকভাবে হেল্প করবে আপনার জাভাস্ক্রিপ্ট এর ক্যারিয়ারে। দেখুন তাহলে আপনি যে আউটপুট টা পেয়েছেন কন্সোলে সেখানে যদি বাম পাশে ত্রিভুজ আকৃতির বাটনে ক্লিক করেন তাহলে অ্যারে টা খুলে যাবে। যেটার ভিতরে অনেক কিছু দেখতে পাবেনঃ



```
> var arrName = ['Rahim', 'Karim', 'Rafiq', 'Jabbar'];  
< undefined  
> arrName  
< ▼ (4) ["Rahim", "Karim", "Rafiq", "Jabbar"] ⓘ  
  0: "Rahim"  
  1: "Karim"  
  2: "Rafiq"  
  3: "Jabbar"  
  length: 4  
  ▶ __proto__: Array(0)  
>
```

দেখুন এখানে প্রত্যেকটা ডাটার সাথে একটা নাম্বার দেখাচ্ছে। হ্যা, এটাকে বলে ইন্ডেক্স নাম্বার। ইন্ডেক্স নাম্বার সাহায্য করবে আপনাকে অ্যারে থেকে স্পেসিফিক কোনো ডাটা বের করে আনতে। এখন জিরো থেকে কেন শুরু হয়েছে? হ্যা প্রোগ্রামিং এর দুনিয়ায় কাউন্ট জিরো থেকেই বেশী শুরু করা হয়। আর

অ্যারে ইন্ডেক্সও জিরো থেকে শুরু হয়। এখন ধরেন আপনি জাব্বার এর নাম অ্যাক্সেস করতে চাচ্ছেন। এটা নরমাল কাউন্টে ৪ নাম্বার পজিশনেও থাকলে যদি জিরো থেকে গণনা শুরু করেন তাহলে এটার পজিশন হচ্ছে ৩ নাম্বারে। আর এই ৩ ই হচ্ছে এটার ইন্ডেক্স নাম্বার। তাে এভাবে অ্যারে থেকে স্পেসিফিক কোনো ভ্যালু অ্যাক্সেস করা হয়ঃ

```
arrName[3]
```

```
< undefined
> arrName
< ▼ (4) ["Rahim", "Karim", "Rafiq", "Jabbar"] ⓘ
  0: "Rahim"
  1: "Karim"
  2: "Rafiq"
  3: "Jabbar"
  length: 4
  ▶ __proto__: Array(0)
> arrName[3]
< "Jabbar"
>
```

এখন যদি আপনি স্পেসিফিক কোনো ভ্যালুও চেঞ্জ করতে চান তাহলেও পারবেন এই ইন্ডেক্স নাম্বারের সাহায্যেইঃ

```
arrName[3] = 'Shafiq';
```

```

< undefined
> arrName
< ▶ (4) ["Rahim", "Karim", "Rafiq", "Jabbar"]
> arrName[3]
< "Jabbar"
> arrName[3] = 'Shafiq'
< "Shafiq"
> arrName[3]
< "Shafiq"
> arrName
< ▶ (4) ["Rahim", "Karim", "Rafiq", "Shafiq"]
> |

```

এ তো গেলো অ্যারে থেকে আইটেম অ্যাক্সেস করা নিয়ে। এবার আসুন অ্যারে যে অবজেক্ট এখন আমি অ্যারে সম্পর্কে কিছু জানতে চাই। কি জানতে চাই? হ্যাঁ আমি অ্যারেতে কয়টা আইটেম আছে জানতে চাই। দেখুন একই জায়গায় নিচে একটা প্রপার্টিজ **length** আছে যেটা জানে যে এই অ্যারেতে কয়টা আইটেম আছেঃ

```

> arrName
< ▼ (4) ["Rahim", "Karim", "Rafiq", "Shafiq"] ⓘ
  0: "Rahim"
  1: "Karim"
  2: "Rafiq"
  3: "Shafiq"
  length: 4
  ▶ __proto__: Array(0)
>

```

এখানে সাধারণত অবজেক্ট এর ভিতরের প্রপার্টিজগুলো অ্যাক্সেস করা হয় **[]** ডট এর মাধ্যমে, একে ডট নোটেশান বলে। আমরা যদি এখন অ্যারেতে কয়টা আইটেম আছে সেটা জানার জন্যে **length** প্রপার্টিজ অ্যাক্সেস করতে চাই তাহলে আমাদের এভাবে লিখতে হবেঃ

```
arrName.length
```

```
> arrName
< ▼ (4) ["Rahim", "Karim", "Rafiq", "Shafiq"] ⓘ
  0: "Rahim"
  1: "Karim"
  2: "Rafiq"
  3: "Shafiq"
  length: 4
  ► __proto__: Array(0)
> arrName.length
< 4
>
```

তো এটা দেখাচ্ছে আসলে এখানে ৪ টা আইটেম আছে। এখন তো আমি ডট নোটেশান এর কথা বললাম। আরো একটা আছে সেটা হচ্ছে ব্র্যাকেটস নোটেশান। এই ডট নোটেশান আর ব্র্যাকেট নোটেশান সব অবজেক্ট এর বেলাতেই ইউজ হয়। তাই এগুলার ইউসেজ মাথায় রাখা অবশ্যই জরুরী। ব্র্যাকেট নোটেশান ইউজ করে এই প্রপার্টিজ অ্যাক্সেস করতে চাইলেঃ

```
arrName['length']
```



```
> var arrName = ['Rahim', 'Karim', 'Rafiq', 'Jabbar'];
< undefined
> arrName
< ▼ (4) ["Rahim", "Karim", "Rafiq", "Jabbar"] ⓘ
  0: "Rahim"
  1: "Karim"
  2: "Rafiq"
  3: "Jabbar"
  length: 4
  __proto__: Array(0)
> arrName['length']
< 4
>
```

নামের সাথে বেশ মিল আছে তাই আশা করি ডট নোটেশান আর ব্র্যাকেট নোটেশান মনে রাখতে হবে না। আপনি কোনটা ইউজ করবেন সেটা আপনার ইচ্ছা, তবে জাভাস্ক্রিপ্ট এর বেলায় ডট নোটেশানটাই বেশী ইউজ করা হয়। মানে এটা কনভেনশন।

এবার কিভাবে অ্যারের সবগুলো আইটেম অ্যাক্সেস করা যেতে পারে? মনে হয় আগের পর্বে লুপ নিয়ে আলোচনা করেছিলাম। একইরকম কাজ বারবার করা লাগলে নাকি লুপ ইউজ করতে হয়। হ্যা! আপনি লুপ ইউজ করে ইজিলি অ্যারের সবগুলো আইটেম অ্যাক্সেস করতে পারবেনঃ

```
for(var i = 0; i < arrName.length; i++) {
  console.log('Name: ' + arrName[i]);
}
```

```

> var arrName = ['Rahim', 'Karim', 'Rafiq', 'Jabbar'];
< undefined
> arrName
< ▶ (4) ["Rahim", "Karim", "Rafiq", "Jabbar"]
> for(var i = 0; i < arrName.length; i++) {
    console.log('Name: ' + arrName[i]);
  }
Name: Rahim
Name: Karim
Name: Rafiq
Name: Jabbar
< undefined
> |

```

এই কোডটা যদি কনফিউজিং হয়ে থাকে তাহলে ভালো করে খেয়াল করুন তাহলেই বুঝতে পারবেন।

এখানে ফর লুপ ইউজ করা হয়েছে এবং প্রত্যেকটা লাইন, এক্সপ্রেশনই নিয়ম মেনে চলছে এবং সবশেষে যেভাবে চাচ্ছেন সেভাবেই সব প্রিন্ট করে দেখাচ্ছে।

এখন যদি নিচের `proto` এটা খুলেন তাহলে বেশ আরো অনেকগুলো প্রপার্টিজ দেখতে পারবেন।

```

> arrName
< ▼ (4) ["Rahim", "Karim", "Rafiq", "Jabbar"] ⓘ
  0: "Rahim"
  1: "Karim"
  2: "Rafiq"
  3: "Jabbar"
  length: 4
  __proto__: Array(0)
    ▶ concat: f concat()
    ▶ constructor: f Array()
    ▶ copyWithin: f copyWithin()
    ▶ entries: f entries()
    ▶ every: f every()
    ▶ fill: f fill()
    ▶ filter: f filter()
    ▶ find: f find()
    ▶ findIndex: f findIndex()
    ▶ forEach: f forEach()
    ▶ includes: f includes()
    ▶ indexOf: f indexOf()
    ▶ join: f join()

```

এখানে কিছু ফাংশন আছে যেগুলো আপনি এই অ্যারের জন্যে ইউজ করতে পারবেন। তবে আমি সবগুলোতে যাবো না। আপনি চাইলে যদি কিউরিসিটি থাকে তাহলে ট্রাই করতে পারেন নিজে থেকে একটা একটা করে। আমি জাভাস্ক্রিপ্ট কে সেজন্যই কমফোর্টেবল ল্যাংগুয়েজ বলি কারণ আপনি ভিতরে সব খুলে দেখতে পারবেন। তাতে সুবিধা হলো আপনি আপনার প্রব্লেম অনেকসময় নিজেই সলভ করে ফেলতে পারবেন। যদি কোনো এরর আসে আপনাকে গুগল করতে হবে না। আপনি জাস্ট ওপেন করে এভাবে দেখতে পারবেন আসলে আমি যেটা ইউজ করতে চাচ্ছি এটা এখানে আছে কিনা বা থাকলেও বানানটা কিরকম। এরকম আরো কতকিছু করতে পারবেন কন্সোল থেকে এভাবে। তবে আমি অ্যারেতে ব্যবহৃত মোস্ট কমন কিছু ফাংশন নিয়ে আলোচনা করবো।

অ্যারেতে শেষের দিক হতে কোনো নতুন আইটেম অ্যাড করতেঃ

একটা অ্যারে যেখানে অলরেডি কিছু আইটেম আছে এখন যদি আরো আইটেম অ্যাড করতে চান তাহলে আপনি শুরুর দিক হতে বা শেষের দিক হতে নতুন আইটেম অ্যাড করতে পারবেন। শেষের দিক হতে আইটেম অ্যাড করতে চাইলেঃ

```
arrName.push('Shafiq');
```



অ্যারেতে শেষের দিক হতে আইটেম রিমুভ করতেঃ

সেইমভাবে আমরা কোনো আইটেম অ্যারে থেকে রিমুভ করতে চাইলে শুরুর দিক থেকে বা শেষের দিক থেকেও করতে পারি। যদি শেষের দিক থেকে করতে চাইঃ

```
arrName.pop();
```

```
> arrName
< ▶ (5) ["Rahim", "Karim", "Rafiq", "Jabbar", "Shafiq"]
> arrName.pop()
< "Shafiq"
> arrName
< ▶ (4) ["Rahim", "Karim", "Rafiq", "Jabbar"]
>
```

অ্যারেতে শুরু দিক হতে আইটেম রিমুভ করতেঃ

সেইমভাবে যদি আমরা শুরু দিক থেকে কোনো আইটেম রিমুভ করতে চাইঃ

```
arrName.shift();
```

```
> arrName
< ▶ (4) ["Rahim", "Karim", "Rafiq", "Jabbar"]
> arrName.shift()
< "Rahim"
> arrName
< ▶ (3) ["Karim", "Rafiq", "Jabbar"]
> |
```

এখন শুরু আইটেম রিমুভ করায় সবার ইন্ডেক্স নাম্বার পরিবর্তন হবে খেয়াল রাখবেন।

অ্যারেতে শুরুর দিক হতে কোনো নতুন আইটেম অ্যাড করতেঃ

সেইমভাবে যদি আমরা শুরুর দিক থেকে কোনো আইটেম অ্যাড করতে চাইঃ

```
arrName.unshift('Shafiq');
```

```
> arrName
< ▶ (3) ["Karim", "Rafiq", "Jabbar"]
> arrName.unshift('Shafiq')
< 4
> arrName
< ▶ (4) ["Shafiq", "Karim", "Rafiq", "Jabbar"]
>
```

এখানেও শুরুর দিকে নতুন আইটেম অ্যাড করায় সবার ইন্ডেক্স নাম্বার চ্যাঞ্জ হবে।

নির্দিষ্ট কোনো আইটেম এর ইন্ডেক্স নাম্বার জানতেঃ

কোনো আইটেম এর যদি ইন্ডেক্স নাম্বার জানতে চান তাহলে কন্সোল থেকে তো জানতে পারবোই, কিন্তু মাঝেমধ্যে কোড থেকেও জানার প্রয়োজন হতে পারে সেক্ষেত্রে আমরা এটা ইউজ করতে পারিঃ

```
arrName.indexOf('Karim');
```

```

> arrName
< ▶ (4) ["Shafiq", "Karim", "Rafiq", "Jabbar"]
> arrName.indexOf('Karim')
< 1
> |

```

ইন্ডেক্স নাম্বার দিয়ে কোনো আইটেম রিমুভ করতেঃ

```
arrName.splice(1, 2);
```

এখানে ইন্ডেক্স নাম্বার কত থেকে শুরু করে কত পর্যন্ত রিমুভ করতে চাচ্ছেন সেগুলো আর্গুমেন্ট হিসেবে এখানে যাবে

```

> var arrName = ['Rahim', 'Karim', 'Rafiq', 'Jabbar'];
< undefined
> arrName
< ▶ (4) ["Rahim", "Karim", "Rafiq", "Jabbar"]
> arrName.splice(1, 2)
< ▶ (2) ["Karim", "Rafiq"]
> arrName
< ▶ (2) ["Rahim", "Jabbar"]
> |

```

আর যদি চান একটা নির্দিষ্ট ইন্ডেক্স নাম্বার থেকে শুরু করে বাকি সবগুলো রিমুভ করতে তাহলে প্রথম আর্গুমেন্টে জাস্ট যেখান থেকে রিমুভ করা শুরু করবেন সে ইন্ডেক্স নাম্বারটা দিলেই হবেঃ

```
arrName.splice(3);
```

```
> var arrName = ['Rahim', 'Karim', 'Rafiq', 'Jabbar'];  
< undefined  
> arrName  
< ▶ (4) ["Rahim", "Karim", "Rafiq", "Jabbar"]  
> arrName.splice(3)  
< ▶ ["Jabbar"]  
> arrName  
< ▶ (3) ["Rahim", "Karim", "Rafiq"]  
> |
```

এখানে একটা জিনিস হয়তো খেয়াল করবেন যে যতবারই আমরা **splice** মেথড ইউজ করতেছি, এটা কিছু ডাটা রিটার্ন করছে। হ্যাঁ, আমরা যেগুলো রিমুভ করতে চাচ্ছি সেগুলোই রিটার্ন করে এই **splice** মেথড। তো আপনি চাইলে রিমুভ করা ডাটাগুলোও আবার আরেকটা ভ্যারিয়েবলে স্টোর করে রাখতে পারবেন। ধরি আমাদের একটা অ্যারে আছেঃ

```
var newArr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 0];
```

এখন এটার ইন্ডেক্স নাম্বার ৩ থেকে বাকী সবগুলো স্প্লাইস করে ফেললাম, আর স্প্লাইসড ডাটাগুলো **spliceArr** তে স্টোর করলামঃ

```
var spliceArr = newArr.splice(3)
```

এখন মেইন অ্যারেটা দেখলেঃ

```
console.log(newArr);
```

```
> console.log(newArr);  
▶ (3) [1, 2, 3]
```

আর আমাদের এই নতুন `spliceArr` টা দেখলেঃ

```
console.log(spliceArr);
```

```
> console.log(spliceArr)  
▶ (7) [4, 5, 6, 7, 8, 9, 0]
```

অ্যারে কপি করতেঃ

যদি একটা অ্যারে থেকে আরেকটা অ্যারে কপি করতে চাই তাহলে অনেকভাবেই করা যায়। সবচেয়ে
সিম্পলভাবে(কপি বাই রেফারেন্স)ঃ

```
var newArr = arrName;
```



```

> var arrName = ['Rahim', 'Karim', 'Rafiq', 'Jabbar'];
< undefined
> var newArr = arrName
< undefined
> newArr
< ▶ (4) ["Rahim", "Karim", "Rafiq", "Jabbar"]
> |

```

অথবা আমরা চাইলে এভাবেও করতে পারি। সেক্ষেত্রে একটু মডিফাই করেও আপনি অ্যারে টাকে কপি করতে পারবেনঃ

```
var newArr = arrName.slice(0, 2);
```

```

> arrName
< ▶ (4) ["Rahim", "Karim", "Rafiq", "Jabbar"]
> var newArr = arrName.slice(0, 2)
< undefined
> newArr
< ▶ (2) ["Rahim", "Karim"]
> |

```

তবে ডিরেক্ট অ্যারে কপি করা আর স্লাইস ইউজ করে কপি করার মধ্যে একটু তফাৎ আছে। ডিরেক্ট অ্যারে কপি করাকে কপি বাই রেফারেন্স বলে। এখানে আমি নতুন যে অ্যারে নিয়েছি সেটা আর আগেরটা সেইম অবজেক্ট কেই রেফার করবে যদি আপনি অ্যাসাইনমেন্ট `=` অপারেটর ইউজ করেন। আপনি যদি যেকোনো একটাতে কিছু চ্যাঞ্জ করেন তাহলে দুইটা তেই সেটার ইফেক্ট পাবেন। কারণ দুইজনই সেইম অবজেক্ট রেফার করতেছে

```
newArr[0] = 'Changed Rahim'
```

```
> var arrName = ['Rahim', 'Karim', 'Rafiq', 'Jabbar'];  
< undefined  
> var newArr = arrName;  
< undefined  
> newArr[0] = 'Changed Rahim'  
< "Changed Rahim"  
> arrName  
< ▶ (4) ["Changed Rahim", "Karim", "Rafiq", "Jabbar"]  
>
```

নতুন অ্যারেতে আইটেম চ্যাঞ্জ করার কারণে সেটা আগের অ্যারেতেও ইফেক্ট পড়েছে। কারণ আসলে দুইটা অ্যারে সেইম মেমোরি লোকেশান বা সেইম অবজেক্টকেই রেফার করছে।

```
arrName === newArr
```

```
> arrName === newArr  
< true
```

আর স্লাইস ইউজ করে কপি করাকে কপি বাই ভ্যালু বলে। এখানে শুধু ভ্যালু বা আইটেমগুলোই কপি হবে নতুন অ্যারেতে। কিন্তু দুইটা অ্যারে সম্পূর্ণ পৃথক পৃথক দুইটা অবজেক্ট

```

> var arrName = ['Rahim', 'Karim', 'Rafiq', 'Jabbar'];
< undefined
> var newArr = arrName.slice()
< undefined
> newArr
< ▶ (4) ["Rahim", "Karim", "Rafiq", "Jabbar"]
> newArr[0] = 'Changed Rahim'
< "Changed Rahim"
> newArr
< ▶ (4) ["Changed Rahim", "Karim", "Rafiq", "Jabbar"]
> arrName
< ▶ (4) ["Rahim", "Karim", "Rafiq", "Jabbar"]
>

```

আর `===` ইউজ করলে দেখবেন ফলস দেখাচ্ছে কারণ দুইটা সেইম অবজেক্ট না, জাস্ট ভ্যালুগুলো কপি হয়েছেঃ

```
arrName === newArr
```

```

> arrName === newArr
< false

```

তো আজকে এই পর্যন্তই। আশা করি অ্যারেটা বুঝতে কোনো সমস্যা হবে না। তারপরেও কোথাও কিছু বুঝতে প্রবলেম হলে আমাকে অবশ্যই জানাবেন। আমি চেষ্টা করবো সব কনফিউসান দূর করে দিতে।
আমাকে সবসময় আমার ইমেইল zonayedpca@gmail.com এ পাবেন

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**