

This app works best with JavaScript enabled.



হাতেকলমে জাভাস্ক্রিপ্ট



জাভাস্ক্রিপ্টঃ স্কোপ(Scope) নিয়ে ধারণা

স্কোপিং(Scoping) জাভাস্ক্রিপ্ট এ আরেকটা গুরুত্বপূর্ণ বিষয়। আপনাকে জানতে হবে আপনার ডিক্লেয়ারকৃত ভ্যারিয়েবল/ফাংশন আপনি কোথায় কোথায় অ্যাক্সেস করতে পারবেন বা কোথায় কোথায় ইউজ করতে পারবেন। কোনো ভ্যারিয়েবল বা ফাংশন প্রাইভেট করতে চাইলে বা কোনো ভ্যারিয়েবলকে সব জায়গা থেকে অ্যাক্সেস করতে চাইলে কিভাবে বা কোথায় সেটাকে ডিক্লেয়ার করতে হবে এসবকিছুই স্কোপিং এর ভিতরে আলোচনা করবো।

জাভাস্ক্রিপ্ট এ প্রধানত দুই ধরনের স্কোপিং হয়ে থাকেঃ

১। লোকাল(Local) স্কোপ

২। গ্লোবাল(Global) স্কোপ

স্কোপিং এ প্রধান আলোচ্য বিষয়গুলো হচ্ছেঃ

- কোথায় কোথায় আপনি আপনার ভ্যারিয়েবল বা ফাংশনকে অ্যাক্সেস করতে পারবেন
- প্রত্যেকটা নতুন ফাংশন একটা স্কোপ তৈরী করে
- লেক্সিকাল স্কোপিং(Lexical Scoping) এ প্যারেন্ট-চাইল্ড রিলেশন
- একটা ফাংশনে ডিক্লেয়ারকৃত ভ্যারিয়েবল আরেকটা ফাংশনে যেতে পারবে না, মানে অ্যাক্সেস করতে পারবেন না।

১। লোকাল(Local) স্কোপঃ সাধারণত ফাংশনের ভিতরে ডিক্লেয়ারকৃত ভ্যারিয়েবলগুলো ঐ ফাংশনের লোকাল ভ্যারিয়েবল। এধরনের ভ্যারিয়েবলগুলোর স্কোপ হচ্ছে লোকাল স্কোপ, অর্থাৎ এই ভ্যারিয়েবলটা শুধুমাত্র ডিক্লেয়ারকৃত ফাংশনের ভিতরেই অ্যাক্সেস করা যাবে। এর বাইরে বা অন্যকোনো ফাংশনে অ্যাক্সেস করা যাবে না।

```
function localScope() {  
    var local = 'I am local to my own function';  
    console.log(local);  
}
```

এটা কল করলে ঠিকঠাক আউটপুট দেখাবেঃ

```
localScope();
```

```

> function localScope() {
    var local = 'I am local to my own function';
    console.log(local);
}
< undefined
> localScope();
    I am local to my own function
< undefined
>

```

কিন্তু এখন যদি আমরা এই `local` ভ্যারিয়েবলটাকে এই `localScope()` এর বাইরে কোথাও বা অন্য কোনো ফাংশনেও অ্যাক্সেস করতে চাই, তাহলে সেটা অ্যাক্সেস করা যাবে না এবং এক্ষেত্রে রিয়েল এরর পাবেন।

```

function localScope() {
    var local = 'I am local to my own function';
    console.log(local);
}

```

```
localScope();
```

```
console.log(local);
```

```
> function localScope() {
    var local = 'I am local to my own function';
    console.log(local);
}
localScope();
console.log(local);
I am local to my own function VM24435:3
✖ ▶ Uncaught ReferenceError: local is not defined VM24435:6
   at <anonymous>:6:13
> |
```

২। গ্লোবাল(Global) স্কোপঃ ভ্যারিয়েবল যদি কোনো ফাংশনের ভিতরে ডিক্লেয়ার না করা হয় তাহলে সেটা গ্লোবাল স্কোপে ডিক্লেয়ারকৃত ভ্যারিয়েবল, মানে ফাংশনের বাইরে যেকোনো জায়গায় ডিক্লেয়ার করা ভ্যারিয়েবলই গ্লোবাল স্কোপ পাবে। এদেরকে গ্লোবাল ভ্যারিয়েবলও বলা হয়। গ্লোবাল ভ্যারিয়েবল যেকোনো ফাংশনের ভিতরে বা যেকোনো জায়গা থেকেই অ্যাক্সেস করতে পারবেন

```
var globalVar = 'I am a Global Variable';
```

```
function globalScope() {
    console.log('Inside a Function: ' + globalVar);
}
```

```
globalScope();
```

```
console.log('Outside: ' + globalVar);
```

```

> var globalVar = 'I am a Global Variable';
  function globalScope() {
    console.log('Inside a Function: ' + globalVar);
  }
  globalScope();
  console.log('Outside: ' + globalVar);
  Inside a Function: I am a Global Variable
  Outside: I am a Global Variable
  < undefined
>

```

এখন স্কোপিং এর এই আইডিয়া যেমন ভ্যারিয়েবলের ক্ষেত্রে প্রযোজ্য, তেমনি ফাংশন এবং অবজেক্ট এর ক্ষেত্রেও প্রযোজ্য।

এর মানে আপনি গ্লোবাল স্কোপে কোনো ফাংশন ডিক্লেয়ার করলে সেটা গ্লোবালি যেকোনো জায়গা থেকেই অ্যাক্সেস করতে পারবেন। উপরে উল্লেখিত সবগুলো উদাহরণে ব্যবহৃত ফাংশনগুলো গ্লোবাল স্কোপে ডিক্লেয়ার করা।

```

function globalFunc() {
  console.log('Global Function');
}

```

এখন এই ফাংশনকে কল করলে ঠিকঠাকভাবে কাজ করবেঃ

```

globalFunc();

```

```

> function globalFunc() {
  console.log('Global Function');
}
< undefined
> globalFunc();
  Global Function
< undefined
. |

```

সেইমভাবে আপনি যদি কোনো ফাংশনের ভিতরে আরেকটা ফাংশন ডিক্লেয়ার করেন তাহলে ফাংশনের ভিতরে ডিক্লেয়ারকৃত ফাংশনটা উক্ত ফাংশনের লোকাল স্কোপ পাবে। এবং এই ফাংশন শুধুমাত্র ঐ ফাংশনের ভিতরেই কল করতে পারবেন। বাইরে এর স্কোপ না থাকায় কল করতে পারবেন না।

```

function anoGlobalFunc() {
  console.log('Global Function');
  function localFunc() {
    console.log('Local Function');
  }
  localFunc();
}

```

লক্ষ্য করুন আমি লোকাল ফাংশন `localFunc()` ঐ ফাংশনের ভিতরেই কল করেছি, কারণ এখানেই এটার স্কোপ আছে। এখন গ্লোবাল ফাংশন `anoGlobalFunc()` কল করলে দুইটাই এক্সিকিউট হবে।

```

anoGlobalFunc();

```

```

> function anoGlobalFunc() {
  console.log('Global Function');
  function localFunc() {
    console.log('Local Function');
  }
  localFunc();
}
< undefined
> anoGlobalFunc()
Global Function
Local Function
< undefined

```

কিন্তু এখন যদি এই লোকাল ফাংশন `localFunc()` আমি `anoGlobalFunc()` এর বাইরে কল করি তাহলে সেটা এরর দেখাবেঃ

```
localFunc();
```

```

> localFunc()
✖ ▶ Uncaught ReferenceError: localFunc is not
  defined
    at <anonymous>:1:1
>

```

সেইম অবজেক্ট এর স্কোপেও। কোনো ফাংশনের ভিতরে ডিক্লেয়ার করা হলে সেটা ঐ ফাংশনের লোকাল স্কোপ পাবে। উক্ত ফাংশনের ভিতরে ছাড়া আর কোথাও অ্যাক্সেস করা যাবে না। আর বাইরে গ্লোবাল স্কোপে ডিক্লেয়ার করা হলে সেটা গ্লোবাল স্কোপ পাবে এবং সব জায়গা থেকে অ্যাক্সেস করা যাবে।

```

var person = {
  name: 'Zonayed Ahmed',

```



```
    age: 21  
}
```

এখন এই অবজেক্ট সব জায়গা থেকেই অ্যাক্সেস করা যাবেঃ

```
console.log(person);
```

```
> var person = {  
    name: 'Zonayed Ahmed',  
    age: 21,  
}  
< undefined  
> console.log(person);  
▶ {name: "Zonayed Ahmed", age: 21}  
< undefined  
>
```

কিন্তু যদি এই অবজেক্ট কোনো ফাংশনের ভিতরে ডিক্লেয়ার করা হয়, তাহলে সেই অবজেক্ট সেই ফাংশনের লোকাল স্কোপ পাবে এবং শুধুমাত্র সেই ফাংশনের ভিতর থেকেই অ্যাক্সেস করা যাবে।

```
function objectFunc() {  
    var localObj = {  
        name: 'Zonayed Ahmed',  
        age: 21  
    }  
    console.log(localObj);  
}
```

```
}
```

এখন এই ফাংশন কল করলে সবকিছু ঠিকঠাক দেখাবেঃ

```
objectFunc();
```

```
> function objectFunc() {  
  var localObj = {  
    name: 'Zonayed Ahmed',  
    age: 21  
  }  
  console.log(localObj);  
}  
↵ undefined  
> objectFunc()  
▶ {name: "Zonayed Ahmed", age: 21}  
↵ undefined  
> |
```

কিন্তু এখন যদি এই অবজেক্ট উক্ত ফাংশনের বাইরে অ্যাক্সেস করতে চাই তাহলে এরর আসবে। কারণ এটা লোকাল স্কোপে ডিক্লেয়ারকৃতঃ

```
console.log(localObj);
```

```
> console.log(localObj);  
✖ ▶ Uncaught ReferenceError: localObj is not  
   defined  
   at <anonymous>:1:13  
>
```

লেক্সিকাল স্কোপিং(Lexical Scoping): লেক্সিকাল স্কোপিং এ একটা প্যারেন্ট ফাংশনের সাথে একটা চাইল্ড ফাংশনের রিলেশন তৈরী হয়। এখন প্যারেন্ট-চাইল্ড ফাংশন মানে কি? হ্যাঁ যদি আমরা একটা ফাংশনের ভিতরে আরেকটা ফাংশন ডিক্লেয়ার করি, তাহলে যে ফাংশনের ভিতরে নতুন ফাংশন ডিক্লেয়ার করলাম সেটা প্যারেন্ট ফাংশন। আর প্যারেন্ট ফাংশনের ভিতরে ডিক্লেয়ারকৃত অন্য যেকোনো ফাংশনই সেই প্যারেন্ট ফাংশনের চাইল্ড ফাংশন। নিচের উদাহরণ থেকে আশা করি ক্লিয়ার ধারণা হবেঃ

```
function parentFunction() {  
    //some codes here  
    function childFunction() {  
        //some more codes  
    }  
}
```

এখন এখানে `parentFunction()` গ্লোবাল ফাংশন আর `childFunction()`

`parentFunction()` এর লোকাল ফাংশন। এখন আমরা জানি কোনো ফাংশনের ভিতরে

ডিক্লেয়ারকৃত ভ্যারিয়েবল, ফাংশন বা অবজেক্ট লোকাল স্কোপ পায় এবং এদেরকে শুধুমাত্র উক্ত ফাংশনের ভিতরেই অ্যাক্সেস করা যায়। কিন্তু লেক্সিকাল স্কোপিং এর কারণে এখানে `parentFunction()` এর ভিতরে ডিক্লেয়ারকৃত লোকাল যেকোনো কিছু `parentFunction()` এর ভিতর থেকে তো অ্যাক্সেস করা যাবেই এবং সেই সাথে এর সব চাইল্ড ফাংশন এর ভিতর থেকেও, এখানে `childFunction()` এর ভিতর থেকেও অ্যাক্সেস করা যাবে।

```
function parentFunction() {  
  var a = 6;  
  function childFunction() {  
    var b = 4;  
    console.log('Sum: ' + (a + b));  
  }  
  childFunction();  
}
```

এখন `parentFunction()` কে কল করলে এটা ঠিক যেসকলভাবে ফলাফল দেখানোর কথা সেসকলভাবেই দেখাবেঃ

```
parentFunction();
```

```
> parentFunction();  
Sum: 10  
< undefined  
> |
```

একইভাবে সেইম কারণে ক্লোজারসও কাজ করে, তবে এটা নিয়ে পরের পর্বে বিস্তারিত থাকবে।

তো আজকে এই পর্যন্তই, ভালো থাকবেন আর পাশের মানুষটিকে ভালো রাখবেন।

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

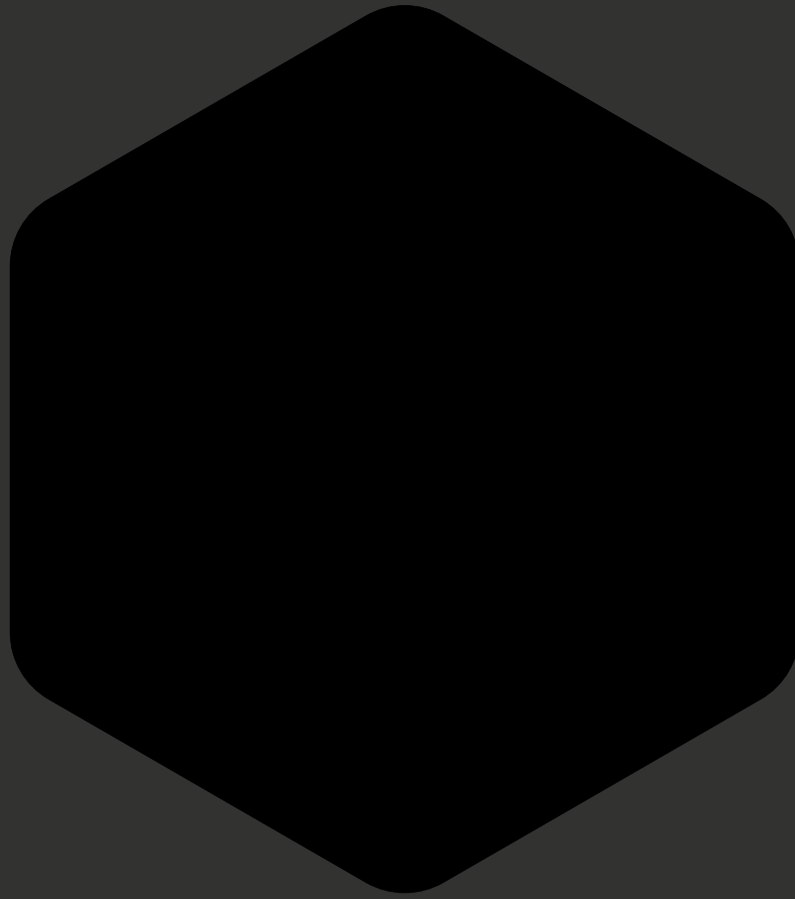
রিসোর্সঃ

এমডিএন ডকুমেন্টেশন

ইউডেমি কোর্স

Eloquent JavaScript

You Don't Know JS



♥ এর সাথে ডেভেলপ করেছে **জুনায়েদ আহমেদ**