

This app works best with JavaScript enabled.



হাতেকলমে জাভাস্ক্রিপ্ট



জাভাস্ক্রিপ্টঃ ব্যাসিক ফাংশন

ধরুন আপনি ২ যোগ ২ এ কতো রেজাল্ট আসে সেটা জানতে চাচ্ছেন। এর জন্যে কয়েক লাইন কোড লিখলেন এবং রেজাল্টও ঠিকঠাক আসলো। এখন আবার পরে ৪ যোগ ৪ এ কতো আসে সেটা জানতে চাইলেন, তো এখন কি করবেন? আবার কিছু লাইন কোড লিখবেন। কিন্তু দেখুন এখানে বারবার আপনি দুইটা ডিজিটের যোগফলই বের করতে চাচ্ছেন। এখন এর জন্যে বারবার নতুন করে কোড না লিখে যদি এমন কোনো কিছুর ব্যবস্থা করা যায় যেটাতে আমি দুইটা নাম্বার দিলেই সেটা আমাকে যোগফল দিয়ে দিবে? হ্যাঁ, প্রোগ্রামিং এ ফাংশনের কাজ সেটাই। আজকে আমি আপনাদের সাথে এই ফাংশন নিয়েই কথা বলবো।

ফাংশনকে একটা ম্যাশিনের সাথে তুলনা করতে পারেন। এখানে আপনি জাস্ট কিছু ইনপুট দিবেন আর সেটা আপনাকে আপনার মতো করে আউটপুট দিবে। আপনার কাছে ধরেন কিছু কাপড় আছে, সেগুলো থেকে আপনি জামা বানাতে চান। আপনি কি করবেন? হয়তো আপনি নিজে বানাতে পারবেন, বা বেশীরভাগ ক্ষেত্রে আমরা যেটা করি কাপড় টা নিয়ে একটা দোকানে দিয়ে আসি আর সেখানে আমাদের শরীরের মাপ রাখা হয়। একটা নির্দিষ্ট সময় পরে আপনি গিয়ে আপনার তৈরী জামা নিয়ে আসেন। এখানে অনেকসময় আপনার জানারও দরকার নাই টেইলর কি করলো আপনার কাপড় দিয়ে। আপনি জাস্ট কাপড় দিলেন, সে আপনার মতো করে জামা তৈরী করে দিলো।

ঠিক ঐরকমভাবেই শুরুতে দেওয়া যোগফলের উদাহরণটা নিলেঃ

```
□□□□□□_□□□□□□□(□□□□□□□, □□□□□□□) {  
    □□□□□□ = □□□□□□□ + □□□□□□□;  
    □□□□□ □□□□□□□;  
}
```

```
□□□□□ □□□□□□□_□□□□□□□□(□, □);  
□□□□□ □□□□□□□_□□□□□□□□(□, □);
```

এখন আমরা জেনেছি জাভাস্ক্রিপ্ট এ বলতে গেলে সবকিছুই অবজেক্ট। তাইলে কি ফাংশনও অবজেক্ট? হ্যাঁ, জাভাস্ক্রিপ্ট এ ফাংশনও অবজেক্ট। জাভাস্ক্রিপ্ট এ ফাংশনকে ফাংশন ক্লাস ফাংশন বলা হয়। কারণ আপনি ফাংশন থেকে ফাংশন রিটার্ন করতে পারবেন। আবার ফাংশনকে ভ্যারিয়েবলেও স্টোর করে রাখতে পারবেন। আবার চাইলে একটা ফাংশনের আর্গুমেন্ট হিসেবে আরেকটা ফাংশনকেও পাস করতে পারবেন। আমি নিচে উদাহরণসহ আলোচনা করবো। সব ক্লিয়ার হয়ে যাবে।

আমরা অনেক বিল্ট-ইন ফাংশন অলরেডি অনেকগুলো ইউজ করেছি ইতিমধ্যেই। যেমনঃ

`console.log('Something here')` আমরা এখানে কন্সোল ডট লগ লিখে ভিতরে একটা কিছু লিখে দিয়েছি আর সেটা ব্রাউজারের কন্সোলে প্রিন্ট হয়ে গিয়েছে। এখন আমাদের আসলে কন্সোল ডট লগ কি কি করে তারপর কিভাবে ব্রাউজারের কন্সোলে পৌঁছে আমার দেওয়া লেখাটাই প্রিন্ট করলো সেটা আমাদের ভাবার দরকার হয়নি। কারণ এটাও একটা ফাংশন। আর এর কাজ হলো আমাদের দেওয়া ইনপুটকে ব্রাউজারের কন্সোলে প্রিন্ট করা, ব্যাস। এরকম আরো অনেক বিল্ট ইন ফাংশন আছে যেগুলো অলরেডি জাভাস্ক্রিপ্ট এর সাথে বা অন্যকিছুর সাথে এসেছে। আপনি জাস্ট কল করলেও কাজ করা শুরু করে দিবে।

এখন এমন পরিস্থিতি আসতে পারে যেখানে আমাদের কাজ বিল্ট-ইন ফাংশনগুলো দিয়ে নাও হতে পারে। সেক্ষেত্রে আমরা নিজেরাই নিজেদের ফাংশন বানাতে পারি। এখন আসি ফাংশন কিভাবে ডিক্লেয়ার করতে হয় সে কথায়। ফাংশন ডিক্লেয়ার করতে অবশ্যই `function` কীওয়ার্ড ইউজ করতে হবে। তারপর ফাংশনটার নাম, তারপর কার্লি ব্র্যাকেট দিয়ে শুরু হবে এবং ভিতরে আপনি কি করবেন সেগুলোর স্টেটমেন্ট যাবে এবং সবশেষে আবার কার্লি ব্র্যাকেট দিয়ে ফাংশন শেষ হয়ে যাবে।

```
function funcName() {  
    //Statements  
}
```

এভাবে ফাংশন ডিক্লেয়ার করাকে ফাংশন স্টেটমেন্ট বলে।

আবার এভাবেও ফাংশন ডিক্লেয়ার করতে পারিঃ

```
var funcName = function() {  
    //Statement
```

```
}
```

এখানে যেকোনো একটা পরিচয়হীন ফাংশন নিয়ে সেটাকে একটা ভ্যারিয়েবলে অ্যাসাইন করেছি। এভাবে ফাংশন ডিক্লেয়ার করাকে ফাংশন এক্সপ্রেশন বলে।

এবার আসি ফাংশন তো ডিক্লেয়ার করা হলো। এখন কিভাবে ইউজ করবো? হ্যাঁ, এবার আপনাকে সেই ফাংশনকে কল বা ডাকতে হবে। না ডাকলে ফাংশন আসবে না বা প্রিন্ট হবে না। ফাংশন এক্সপ্রেশন বা স্টেটমেন্ট যাই হউক না কেন ফাংশন কল উপরের উদাহরণগুলোর ক্ষেত্রেঃ

```
funcName();
```

জাস্ট এভাবেই শেষে প্রথম ব্র্যাকেটসগুলো দিলেই ফাংশন সেটা কল হয়ে যাবে।

```
function funcName() {  
    console.log('Hello I am from the function');  
}
```

```
funcName();
```

এবং

```
var funcName = function() {  
    console.log('Hello I am from the function');  
}
```

```
funcName();
```

উপরের দুইটা কোডেরই আউটপুট সেইম আসবেঃ

```
    Hello I am from the function  
    > |
```

Output

এখানে ফাংশন জাস্ট নরমালি ভিতরে থাকা কন্সোল ডট লগ দেখিয়ে যাচ্ছে। এটা তেমন ইউজফুল মনে হচ্ছে না। আমরা চাচ্ছি এমন একটা ফাংশন বানাতে যেটা দুইটা সংখ্যা নিবে এবং তাদের যোগফল রিটার্ন করে দিবে।

```
function aFunc(parameters) {  
    //do something with the parameters  
    //return the result  
}
```

```
aFunc(arguments);
```

হ্যাঁ, এখানেই আসে প্যারামিটার আর আর্গুমেন্ট এর কথা। আমরা ফাংশনের ভিতরে ডাটা দিতে এগুলো ইউজ করি। এখন ফাংশন যত ইচ্ছা ততোটাই প্যারামিটার নিতে পারবে। দুইটা সংখ্যা না, চাইলে একশো সংখ্যাও নিতে পারবে। কিন্তু যতটা প্যারামিটার দিবেন আপনার ফাংশনে, ঠিক ততোটাই আর্গুমেন্ট দিতে হবে ফাংশন কল করার সময়। নতুবা আপনার ফাংশন ঠিকমতো রেজাল্ট দেখাবে না। প্যারামিটার আর আর্গুমেন্ট নিয়ে প্যাঁচ লেগে গেলে নিচের উদাহরণ দেখে ক্লিয়ার হবেন। ফাংশন ডিক্লেয়ার করার সময় আমরা প্রথম ব্র্যাকেটস এর ভিতরে প্যারামিটার পাস করেছি আর কল করার সময় আর্গুমেন্টগুলো পাস করেছিঃ

```
function sumMachine(a, b) {  
    var sum = a + b;  
    return sum;  
}
```

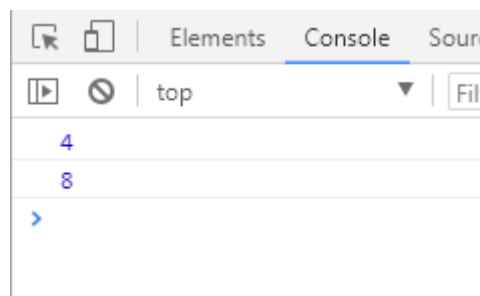
```
console.log(sumMachine(2, 2));  
console.log(sumMachine(4, 4));
```

এখানে আমি ফাংশন স্টেটমেন্ট ইউজ করেছি, আপনি চাইলে এক্সপ্রেশনও ইউজ করতে পারবেন। স্টেটমেন্ট আর এক্সপ্রেশন এর ব্যাপারে ডিটেইলস অন্য পর্বে লিখবো। এখন শুধু জেনে রাখুন কোনটা স্টেটমেন্ট আর কোনটা এক্সপ্রেশন।

এখানে রিটার্ন স্টেটমেন্ট খেয়াল করবেন। আমরা উপরে যে ফাংশন ইউজ করে কন্সোলে কিছু লগ করিয়েছি সেটা কিন্তু কোনো ভ্যালু প্রডিউস করে নি। কারণ সেই ফাংশন থেকে কিছু রিটার্ন করা হয়নি। আপনি আপনার ফাংশন থেকে কিছু একটা ভ্যালু ফিরে পেতে চাইলে রিটার্ন স্টেটমেন্ট ইউজ করতে হবে। আপনি ভ্যারিয়েবল বা যেকোনো ডাটা টাইপের ডাটাই এখানে রিটার্ন করতে পারবেন। আর ফাংশনের

ভিতরে শুধু একটা রিটার্ন স্টেটমেন্টই যাবে। দুইটা ডিক্লেয়ার করলেও প্রথমটা ইউজ করে রিটার্ন করে ফাংশন তার কাজ করা বন্ধ করে দিবে। তাই পরেরটা কখনোই কাজ করবে না।

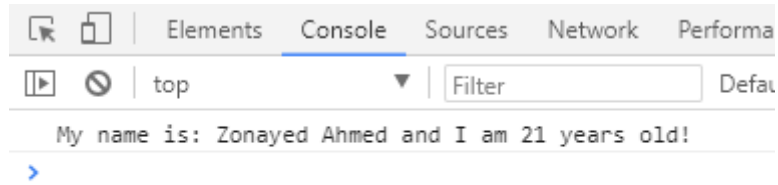
এখানে `sumMachine` নামে একটা ফাংশন দুইটা প্যারামিটার `a` এবং `b` নিয়েছে। ফাংশনের ভিতরে দুইটা যোগ করে একটা ভ্যারিয়েবল `sum` এ নেওয়া হয়েছে। তারপর আমরা `sum` রিটার্ন করেছি। এখানে দেখুন আমাদের `sum` কে কোথাও প্রিন্ট করতে বলা হয় নি। আমি ফাংশনটাকে কল করে সেটা প্রিন্ট করতে বলেছি। এখানে `sum` রিটার্ন করায় ফাংশনের ভ্যালু হিসেবে `sum` টাই প্রিন্ট হচ্ছে কন্সোলেঃ



এখান, আপনি আর্গুমেন্ট হিসেবে যেকোনো ভ্যারিয়েবল, স্ট্রিং, বুলিয়ান, অ্যারে বা অবজেক্ট যেকোনো ডাটা টাইপই পাস করতে পারবেন।

```
function printMyName(name, age) {  
    console.log('My name is: ' + name + ' and I am ' +  
age + ' years old!');  
}
```

```
printMyName('Zonayed Ahmed', 21);
```

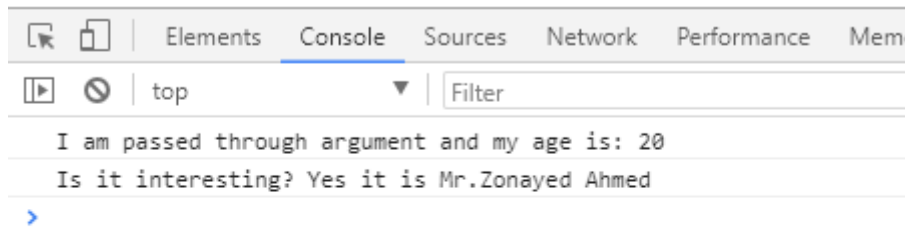
এখন জাভাস্ক্রিপ্ট এ যেহেতু ফাংশনকে ফাস্ট ক্লাশ ফাংশন ধরা হয় তাই আপনি চাইলে একটা ফাংশনকে আরেকটা ফাংশনের আর্গুমেন্ট হিসেবে পাস করতে পারবেন। নিচের কোডটা ভালো করে খেয়াল করুনঃ

```
function callMyName(name, callback) {  
    var myAge = 20;  
    callback(myAge);  
    console.log('Is it interesting? Yes it is Mr.' +  
name);  
}
```

```
function hello(age) {  
    console.log('I am passed through argument and my age  
is: ' + age);  
}
```

```
callMyName('Zonayed Ahmed', hello);
```

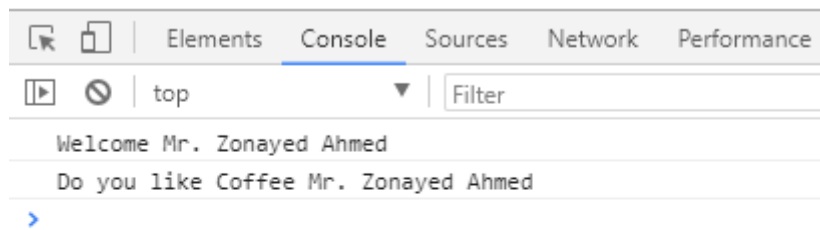
আউটপুটঃ



সেইম কারণে আপনি একটা ফাংশন থেকে আরেকটা ফাংশনও রিটার্ন করতে পারবেনঃ

```
function welcomeMsg(name) {  
    console.log('Welcome Mr. ' + name);  
    return function options(menu) {  
        console.log('Do you like ' + menu + ' Mr. ' +  
name);  
    }  
}
```

```
welcomeMsg('Zonayed Ahmed')('Coffee');
```

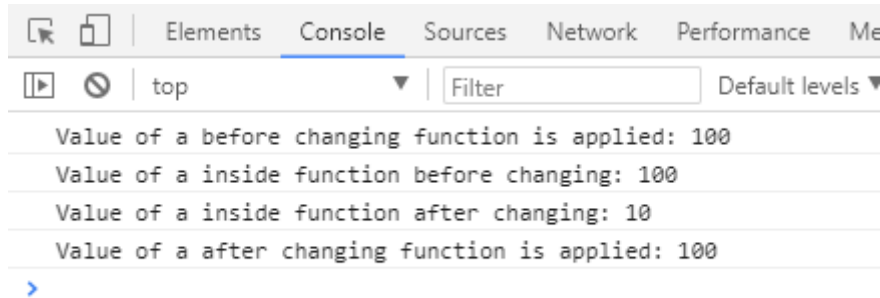


এখন অনেকের মনে একটা প্রশ্ন থেকে থাকতে পারে। যেমন এই প্রোগ্রামটা লক্ষ্য করুনঃ

```
function changeValue(a) {  
    console.log('Value of a inside function before  
changing: ' + a);  
    a = 10;  
    console.log('Value of a inside function after  
changing: ' + a);  
    return a;  
}
```

```
var a = 100;  
console.log('Value of a before changing function is  
applied: ' + a);  
changeValue(a);  
console.log('Value of a after changing function is  
applied: ' + a);
```

এখন এখানে আমরা `a` কে আর্গুমেন্ট হিসাবে ফাংশনে পাস করে সেটার ভ্যালু চ্যঞ্জ করে রিটার্ন করে আবার প্রিন্ট করেছি। কি মনে হচ্ছে ফাংশন অ্যাপ্লাই করার পর কি `a` এর ভ্যালু চ্যঞ্জ হবে? আউটপুট দেখুন তাইলেঃ



দেখুন `a` এর ভ্যালু চ্যঞ্জ হয়নি। কারণ এখানে আর্গুমেন্ট হিসাবে `a` পাস করা হয়েছে জাস্ট ভ্যালু হিসাবে। মানে জাস্ট আপনি `a` এর একটা কপি পাঠিয়েছেন। আসল `a` বা রেফারেন্সটা পাঠাননি। তাই আপনার অরিজিনাল ভ্যালু চ্যঞ্জ হয়নি। এটা থেকে বুঝা গেলো আপনি এভাবে আর্গুমেন্ট পাস করলে সেটার একটা কপি পাস হয়, রেফারেন্স পাস হয় না।

প্রশ্নোত্তরঃ

১। কোনটা রেফারেন্স আর কোনটা বাই ভ্যালু হিসাবে পাস করেছি কিভাবে বুঝবো?

উঃ জাভাস্ক্রিপ্ট এর ফাংশনে যখন আপনি কোনো(প্রিমিটিভ ডাটা টাইপ) আর্গুমেন্ট পাস করবেন ঐটা সবসময় ভ্যালু হিসাবে পাস হবে। তবে আপনি যদি নন-প্রিমিটিভ ডাটা টাইপ(বা আরেকনামে রেফারেন্স ডাটা টাইপ) যেমনঃ অবজেক্ট, অ্যারে পাস করে এতে কোনো পরিবর্তন করেন(পুশ, বা অবজেক্ট এর প্রপার্টি) তাহলে সেটা বাই রেফারেন্স পাস হবে(যেহেতু এরা রেফারেন্স ডাটা টাইপ) এবং আপনার অরিজিনাল ডাটাতে সেটার ইফেক্ট পড়বে:

```
> function aFunc(a) {  
  console.log('Before changing the property: ' + a.one);  
  a.one = 'two';  
  console.log('After changing the property: ' + a.one);  
}  
  
var a = {  
  one: 'one'  
}  
  
aFunc(a)  
Before changing the property: one  
After changing the property: two  
< undefined  
> a  
< ▶ {one: "two"}  
> |
```

অবজেক্ট রেফারেন্স ডাটা টাইপ হওয়ায় ফাংশন আর্গুমেন্ট হিসাবে পাস করে চ্যাঞ্জ করার পরও সেটার ইফেক্ট মেইন অবজেক্ট এ পড়েছে

আজকে এই পর্যন্তই। ফাংশনটা প্রচুর প্র্যাক্টিস করতে হবে নাইলে আয়ত্তে আনা যাবে না। তাই আশা করি আপনি আপনার মতো করে বিভিন্নরকম ফাংশন প্র্যাক্টিস করবেন [?]

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

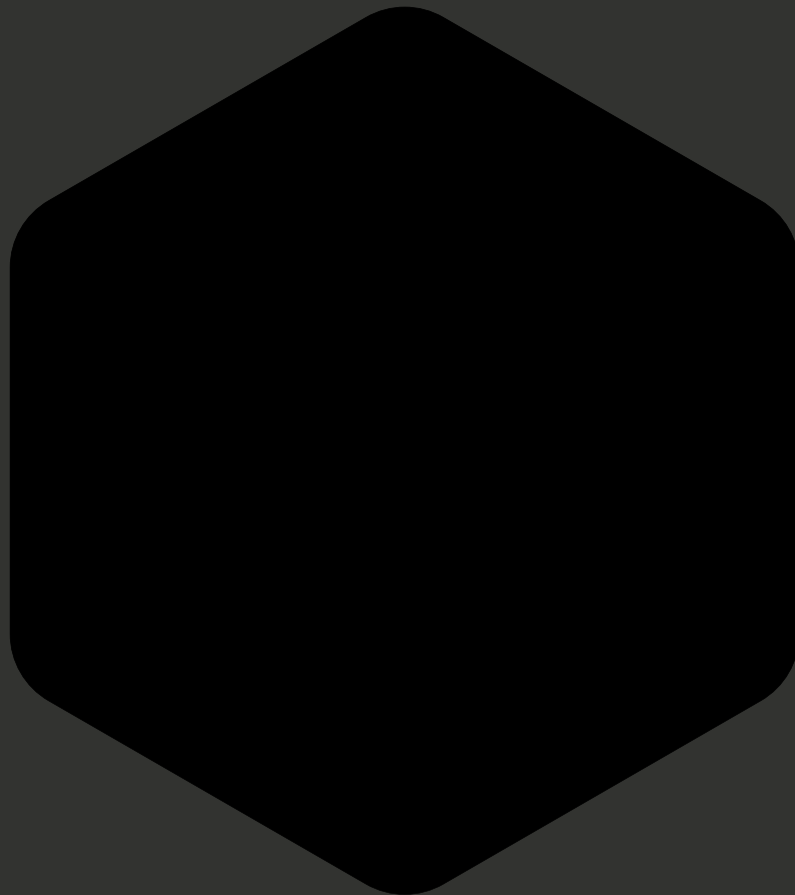
রিসোর্সঃ

এমডিএন ডকুমেন্টেশন

ইউডেমি কোর্স

Eloquent JavaScript

You Don't Know JS



♥ এর সাথে ডেভেলপ করেছে **জুনায়েদ আহমেদ**