

This app works best with JavaScript enabled.



হাতেকলমে জাভাস্ক্রিপ্ট



## জাভাস্ক্রিপ্টঃ বিহাইন্ড দ্যা সীন

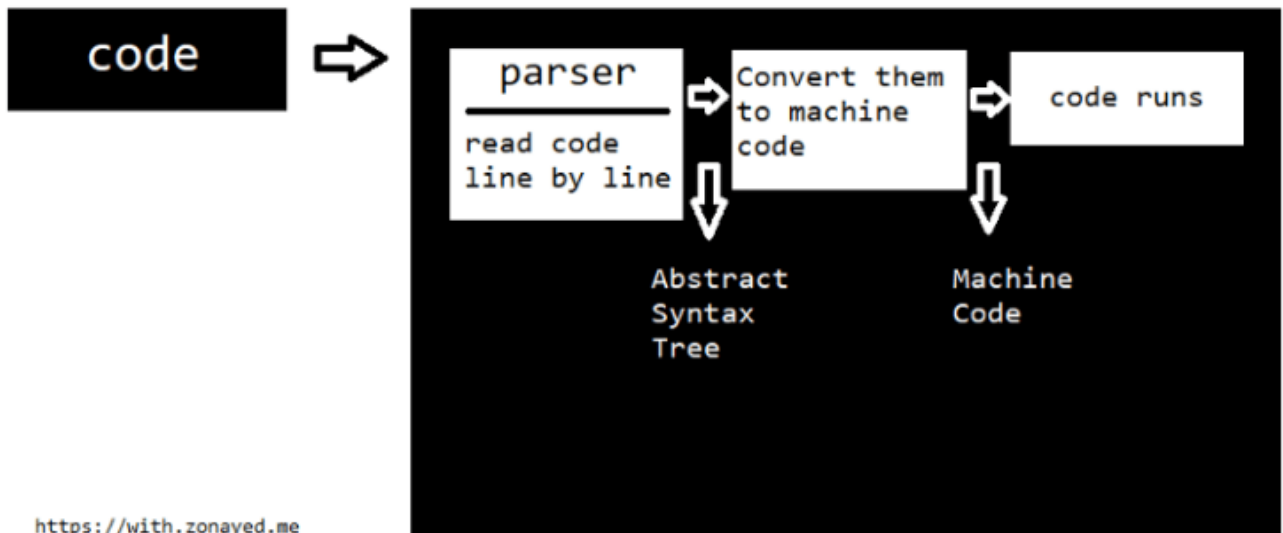
কোড করা অনেক সহজ। কিন্তু ভালো কোড লিখা আর সেই কোড কিভাবে কাজ করছে সেটা বুঝা অনেক কঠিন। কঠিন বলে ডিমোন্স্ট্রেট করতে চাই না। আসলে কোড কিভাবে কাজ করছে সেটা হচ্ছে বুঝার বিষয়। একবার যদি বুঝে যান কোড বিহাইন্ড দ্যা সীনে কিভাবে কাজ করছে তাইলে আপনি যেকোনোভাবেই আপনার সুবিধামতো কোড লিখে যেতে পারবেন। আমার আজকের এই পর্বে তাই জাভাস্ক্রিপ্ট বিহাইন্ড দ্যা সীনে কিভাবে কাজ করে সেগুলো নিয়েই যতটুকু সম্ভব বিস্তারিত আলোচনা করবো।

জাভাস্ক্রিপ্ট সাধারণত কোনো এনভারোমেন্ট এ হোস্ট করা থাকে। প্রায়সময় সেটা আমাদের বহুল ব্যবহৃত

ব্রাউজার হতে পারে, বা অন্য কোনো অ্যাপ্লিকেশন প্রোগ্রাম যেমন নোড জেএসও হতে পারে।

এখন এই জাভাস্ক্রিপ্ট যেখানে হোস্ট করা থাকে, সেখানে একটা জাভাস্ক্রিপ্ট এর ইঞ্জিনও(Google V8 Engine, Spider Monkey, JavaScript Core etc...) থাকে যার কাজ হচ্ছে এই জাভাস্ক্রিপ্ট কোডকে এক্সিকিউট করা। কোথায় কোন ইঞ্জিন থাকে সেটা পূর্বে আলোচনা করা হয়েছে।

এখন এই জাভাস্ক্রিপ্ট কোড কয়েকটা ধাপে এক্সিকিউট হয়ে রান করে। আমরা নিচের ডায়াগ্রাম থেকে সেটা দেখতে পারিঃ



ফটোশপটা ক্রাশ করতেছিলো কোনো কারনে তাই ভালোভাবে করতে পারিনাই। পরে আপডেট করে দিবো।

এখানে কোড প্রথমে পার্সারের মধ্যে দিয়ে যায়। এখানে সাধারণত সিনট্যাক্স এরর চ্যাক করা হয়। যদি কোনো সিনট্যাক্স এরর পাওয়া যায় তাহলে এখান থেকে সে এররটা থ্রো করা হয়। আর যদি সব সিনট্যাক্স কারেক্ট হয়ে থাকে, তাহলে এখান থেকে অ্যাবস্ট্রাক্ট সিনট্যাক্স ট্রি(Abstract Syntax Tree) নামে একটা ডাটা স্ট্রাকচার তৈরী করা হয়।

তারপরের ধাপে এই কোড ম্যাশিন কোডে কনভার্ট করা হয়, যেটা আসলে আমাদের কম্পিউটার বুঝতে পারে। এবং সবশেষে আমাদের কোড রান হয়।

এখন কেন এতো ধাপ ট্রান্স করতে হয় একটা কোড রান করার আগে? হ্যাঁ আমাদের কম্পিউটার ব্যাসিক্যালি ম্যাশিন কোড ছাড়া অন্যকিছু বুঝতে পারে না। আমরা যে ল্যাংগুয়েজেই কোড লিখি না কেন সেটা অবশ্যই ম্যাশিন কোডে কনভার্ট হতে হয়, নতুবা আমাদের কম্পিউটার সেটা বুঝতে পারে না।

এক্সিকিউশন কন্টেক্সট এবং এক্সিকিউশন স্ট্যাক

এক্সিকিউশন কন্টেক্সট(Execution Context) এবং এক্সিকিউশন স্ট্যাক(Execution Stack) সম্পর্কে পরিষ্কার ধারণা থাকা খুবই গুরুত্বপূর্ণ। জাভাস্ক্রিপ্ট এ অনেক ট্রিকি বিষয় রয়েছে যেগুলো বুঝা অনেক কষ্টসাধ্য যদি না আপনি ঠিক মতো এই এক্সিকিউশন কন্টেক্সট(Execution Context) আর এক্সিকিউশন স্ট্যাক(Execution Stack) বুঝতে পারেন।

এক্সিকিউশন কন্টেক্সট(Execution Context):

বাই ডিফল্ট জাভাস্ক্রিপ্ট গ্লোবাল কন্টেক্সট এক্সিকিউট করে। অন্যান্য কিছু ল্যাঙ্গুয়েজের মতো কোনো স্পেসিফিক ফাংশন থেকে এক্সিকিউট হয় না। আর জাভাস্ক্রিপ্ট এর গ্লোবাল এক্সিকিউশন কন্টেক্সট হচ্ছে ব্রাউজারের `window` অবজেক্ট। কোনো গ্লোবাল ভ্যারিয়েবল ডিক্লেয়ার করা বা `window` অবজেক্ট এ অ্যাসাইন করা একই কথা

```
var a = 10;
```

এখন যদি দেখি `window` এর ভিতরে `a` আছে কিনা তাহলে দেখতে পারঃ

```
window.a
```

তাহলে দেখবেন যে এটা সেইম আপনার ডিক্লেয়ারকৃত `a` এর ভ্যালুই দেখাচ্ছে

```
> a
< 10
> window.a
< 10
> |
```

```
a === window.a //true
```

মানে দুইটা একই জিনিস। তো গ্লোবাল ভ্যারিয়েবল ডিক্লেয়ার করা বা `window` অবজেক্ট এ অ্যাসাইন করা একই কথা।

তবে আপনার এনভারোমেন্ট যদি ব্রাউজার না হয় তাহলে এই গ্লোবাল এক্সিকিউশন কন্টেক্সট এনভারোমেন্ট অনুযায়ী ভিন্ন ভিন্ন হতে পারে। যেমন নোড জেএস এর গ্লোবাল এক্সিকিউশন কন্টেক্সট হচ্ছে `global` অবজেক্ট। এনভারোমেন্ট অনুযায়ী এটা চ্যাঞ্জ হতে পারে।

এক্সিকিউশন কন্টেক্সট নিয়ে পরে আরো বিস্তারিত আলোচনা করা হয়েছে।

এক্সিকিউশন স্ট্যাক(Execution Stack):

জাভাস্ক্রিপ্ট এ এক্সিকিউশন স্ট্যাক(Execution Stack) সম্পর্কে ক্লিয়ার ধারণার জন্যে আমরা একটা উদাহরণ ব্যবহার করবো এবং দেখবো সেই কোডগুলো কিভাবে এক্সিকিউট হচ্ছে।

```
var name = 'Zonayed';
```

```
function first() {  
    var welcome = 'Hello ';  
    second();  
    console.log(welcome + name);  
}
```

```
function second() {  
    var welcome2 = 'Hi! ';  
    third();  
    console.log(welcome2 + name);  
}
```

```
function third() {  
    var welcome3 = 'Hey! ';  
    console.log(welcome3 + name);  
}
```

```
first();
```

উপরের এই কোড রান করলে দেখবেন এভাবে রান হয়েছেঃ

Hey! Zonayed
Hi! Zonayed
Hello Zonayed
< undefined

শেষেরগুলো আগে আসতেছে আর আগেরগুলো পরে। কেমন এলোমেলো লাগতেছে না? হ্যাঁ এখানেই আমাদের এক্সিকিউশন স্ট্যাকটা কিভাবে কাজ করতে সেটা বুঝতে হবে। তাইলেই একমাত্র তখন আমাদের পরিষ্কার ধারণা হবে কিভাবে এই কোড রান করলো। প্রথমে আমার নিচের এই ডায়াগ্রামটা লক্ষ্য করুনঃ

```

var name = 'Zonayed'; _____ 1

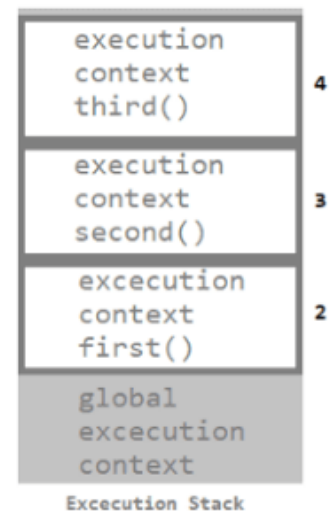
function first() { _____ 2
  var welcome = 'Hello '; _____ 6
  second(); _____ 7
  console.log(welcome + name); _____ 13
}

function second() { _____ 3
  var welcome2 = 'Hi! '; _____ 8
  third(); _____ 9
  console.log(welcome2 + name); _____ 12
}

function third() { _____ 4
  var welcome3 = 'Hey! '; _____ 10
  console.log(welcome3 + name); _____ 11
}

first(); _____ 5

```



<https://with.zonayed.me>

প্রয়োজনে জুম করে দেখুন

এখানে নাম্বার কোডগুলো সিরিয়াল নাম্বার অনুযায়ী এক্সিকিউট হবে। আর ডান পাশের ডায়াগ্রামটা ব্যাসিকালি এক্সিকিউশন স্ট্যাক। স্ট্যাক ডাটা স্ট্রাকচার সম্পর্কে যদি আপনার ধারণা না থেকে থাকে তাহলে স্ট্যাক এমন একটা স্ট্রাকচার যেখানে ফাস্ট ইন লাস্ট আউট সিস্টেম মেইন্টেইন করা হয়। এর মানে হচ্ছে

এখানে ডাটা বের হওয়ার একটাই মাত্র রাস্তা থাকবে, আর যখনই কোনো ডাটা বের করা হবে সেটা একদম উপর থেকে বের করা যাবে। আপনি যদি অনেকগুলো কাঁচের প্লেট একটার উপরে আরেকটা স্ট্যাক দিয়ে রাখেন, তাহলে আপনি কোনো একটা প্লেট নিতে চাইলে প্রথমে আপনাকে উপরের প্লেটটাই নিতে হবে, নাইলে নিচে থেকে নিতে পারবেন না বা নিতে চাইলেও দুর্ঘটনার সম্ভাবনা থাকবে। আপনি সবার শেষে যে প্লেটটা রাখবেন স্ট্যাক ডাটা স্ট্রাকচার সিস্টেমে আপনি কোনো প্লেট নিতে চাইলে সেটাই নিতে হবে যেহেতু এই প্লেটটাই আপনার সবার উপরে। আর সবার প্রথমে যে প্লেটটা রেখেছেন সেটা হবে সবার নিচে, উপরের সবগুলো একটা একটা করে নেওয়া শেষ হলে পরেই আপনি সেই প্লেটটা নিতে পারবেন।

এখন আমাদের কোড যখন নাম্বার ১ এ তখন `name` ভ্যারিয়েবলে `Zonayed` অ্যাসাইন হয়ে যায়। তারপর শুধুমাত্র ফাংশন ডেফিনেশনগুলো একটা একটা করে প্রথমে `first()`, তারপরে `second()` এবং সবশেষে `third()` এক্সিকিউশন স্ট্যাক এ গ্লোবাল এক্সিকিউশন কন্ট্রোল এ চলে যায়। এখানে ফাংশনের ভিতরে থেকে কিছুই এক্সিকিউট হবে না যেহেতু আমরা ফাংশন কোনোটাই এখন পর্যন্ত কল করি নাই।

তারপরের নাম্বার ৫ এ আমরা `first()` ফাংশনটাকে কল করেছি। এখন এই ফাংশ্ট এক্সিকিউশন স্ট্যাক এ গ্লোবাল এক্সিকিউশন কন্ট্রোল এর উপরে স্থান পাবে এবং এখন `first()` এর ভিতর থেকে কোড এক্সিকিউট হওয়া শুরু করবে। এখন নাম্বার ৬ এ একটা ভ্যারিয়েবল নিয়ে সেখানে কিছু ডাটা অ্যাসাইন করা হলো। কিন্তু আবার নাম্বার ৭ এ আরেকটা ফাংশন `second()` কে কল করা হয়েছে। তাই এখন এক্সিকিউশন স্ট্যাক এ `first()` এর উপরে নতুন আরেকটা এক্সিকিউশন কন্ট্রোল `second()` স্থান নিবে এবং এখন `second()` এর ভিতর থেকে কোড এক্সিকিউট হওয়া শুরু করবে। এখন এখানেও নাম্বার ৮ এ নতুন একটা ভ্যারিয়েবলে কিছু ডাটা অ্যাসাইন করা হয়েছে। আবার এর পরের নাম্বার ৯ এ আবার আরেকটা ফাংশন `third()` কে কল করা হয়েছে। তাই এখন এক্সিকিউশন স্ট্যাক এ নতুন আরেকটা এক্সিকিউশন কন্ট্রোল `third()` স্থান পাবে এবং এখন `third()` এর ভিতর থেকে কোড এক্সিকিউট হওয়া শুরু করবে। এখন `third()` এর ভিতরে নাম্বার ১০ এ আবার নতুন একটা ভ্যারিয়েবল নিয়ে কিছু ডাটা অ্যাসাইন করা হয়েছে। এবং পরের লাইনে কন্সোল লগ করা হয়েছে যেটা



এখন আপনার কন্সোলে শো করবে। তাই এখানকার লগটাই আপনি প্রথমে দেখতে পাবেন আপনার আউটপুটে। এখন `third()` এক্সিকিউট হওয়া শেষ এবং এটা এখান থেকে সরে যাবে। তাই এখন এক্সিকিউশন কন্টেক্সট যাবে `second()` এ যেহেতু এটাই এখন এক্সিকিউশন স্ট্যাক এর একদম উপরে। এখন `second()` এর ভিতরে নাম্বার ১২ এর কন্সোল লগটা প্রিন্ট হবে আউটপুটে এবং তাই এটাই আপনি দ্বিতীয় পজিশনে আউটপুট হিসাবে দেখতে পাবেন। এখন সেইমভাবে `second()` ও শেষ এবং এক্সিকিউশন স্ট্যাক থেকে চলে যাবে। এখন এক্সিকিউশন স্ট্যাক এ একদম উপরে থাকবে `first()`। তাই এখন `first()` এক্সিকিউট হবে। `first()` এ নাম্বার ১৩ তে থাকা কন্সোল লগ প্রিন্ট হবে আউটপুটে এবং এই ফাংশন ফিনিশ হয়ে যাবে এবং এক্সিকিউশন স্ট্যাক থেকে সরে যাবে। আর তাই এখানকার আউটপুটটা সবার শেষে দেখতে পাবেন। এভাবেই জাভাস্ক্রিপ্ট এ এক্সিকিউশন স্ট্যাক কাজ করে। আর তাই আমরা এরকম আউটপুট দেখতে পাই।

এক্সিকিউশন কন্টেক্সট (Execution Context) নিয়ে বিস্তারিতঃ

এক্সিকিউশন কন্টেক্সট নিয়ে বিস্তারিত জানা খুবই গুরুত্বপূর্ণ। আপনি হয়তো জাভাস্ক্রিপ্ট এ দেখে থাকবেন আপনি চাইলে একটা ফাংশন ডিফাইন করার আগেই কল করতে পারবেন এবং সেই কোড কমপ্লিটলি ঠিকঠাক কাজ করবে।

```
first();
```

```
function first() {  
  console.log('Jello');  
}
```

এই কোড পুরোটা একসাথে রান করলে ঠিকঠাক কাজ করবে।

অথবা একটা ভ্যারিয়েবলও ইউজ করতে পারবেন এটা ডিফাইন করার আগে(ডিফাইন করার আগে ইউজ করলে সেটার ভ্যালু আন্ডিফাইন্ড দেখাবে, কোনো এরর দেখাবে না)।

```
console.log(welcome);
```

```
var welcome = 'Hello World!';
```

আউটপুট আসবে এরকম কিন্তু কোনো এরর আসবে নাঃ

```
> console.log(welcome);  
var welcome = 'Hello World!';  
undefined  
← undefined  
>
```

কি কারনে এরকম হয়? হ্যাঁ সেজন্যেই আমাদের এক্সিকিউশন কন্ট্রোল নিয়ে বিস্তারিত গভীর নলেজ থাকতে হবে তাহলেই আপনি বুঝতে পারবেন এগুলো কেন কাজ করে।

যখন একটা ফাংশন কল করা হয়, তখন এখানে দুইটা ফেজ ঘটেঃ

১। ক্রিয়েশন ফেজ

২। এক্সিকিউশন ফেজ

১। ক্রিয়েশন ফেজঃ ক্রিয়েশন ফেজে প্রথমে ভ্যারিয়েবল অবজেক্ট ক্রিয়েট করা হয়। তারপর স্কোপ চেইন ক্রিয়েট করা হয়। স্কোপ নিয়ে পরের পর্বে আলোচনা করা হয়েছে। তারপর সবশেষে **this** কীওয়ার্ড এর

ভ্যালু ডিটারমাইন করা হয়। `this` যদি চিনে না থাকেন তাহলে চিন্তার কারণ নেই। আপাতত মাথায় রাখবেন এখানেই এটার ভ্যালু ডিটারমাইন করা হয়, আমি পরে `this` নিয়ে আলোচনা করবো।

সোজা বাংলায় ক্রিয়েশন ফেজে কি কি হয় বললেঃ

- ফাংশনে যতগুলো আর্গুমেন্ট পাস করা হয়েছে সেগুলোর আর্গুমেন্ট লিস্ট তৈরী করা হয়
- কোড সব ফাংশন স্ক্যান করে এবং প্রত্যেকটা ফাংশন একটা ভ্যারিয়েবল অবজেক্ট এ স্টোর করা হয় যেটা সাধারণত ফাংশনটাকে পয়েন্ট করে।
- তারপর কোড ভ্যারিয়েবল এর ডিক্লেয়ারেশনগুলো খোঁজ করে। এবং প্রত্যেকটা ডিক্লেয়ারকৃত ভ্যারিয়েবলের জন্যে `praparti` সেট করে দেয়।

২। এক্সিকিউশন ফেজঃ তারপর এই ফেজে কোড এক্সিকিউট হয়। কোড এক্সিকিউশনের ক্ষেত্রে এক্সিকিউশন কন্ট্রোল মেইন্টেইন করে কোড লাইন বাই লাইন এক্সিকিউট হয়। এজন্যেই পরে ফাংশন ডিফাইন করার আগে কল করলেও সেটা ঠিকঠাক কাজ করে। আর কোনো ভ্যারিয়েবল ডিক্লেয়ার করার আগে ইউজ করলে সেটার ভ্যালু `undefined` হিসাবে সেট হয়ে যায় এবং কোনো এরর শো করে না।

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

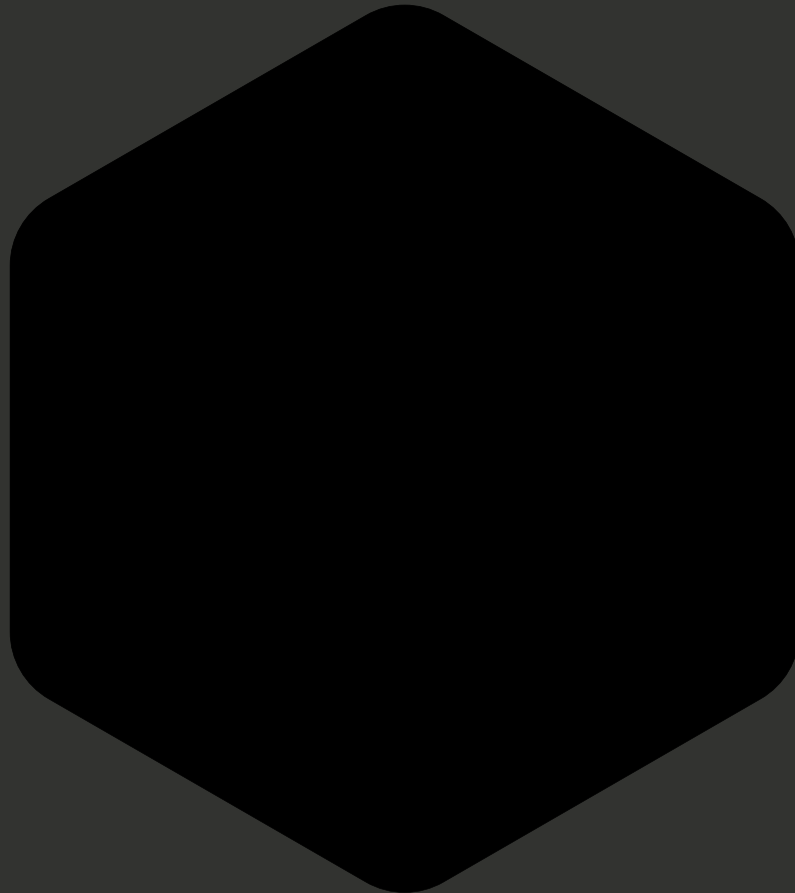
রিসোর্সঃ

এমডিএন ডকুমেন্টেশন

ইউডেমি কোর্স

Eloquent JavaScript

You Don't Know JS



♥ এর সাথে ডেভেলপ করেছে **জুনায়েদ আহমেদ**