

This app works best with JavaScript enabled.



হাতেকলমে জাভাস্ক্রিপ্ট



# জাভাস্ক্রিপ্ট অ্যাডভান্সঃ ইনহেরিট্যান্স (Inheritance)

অবজেক্ট ওরিয়েন্টেড প্রোগ্রামিং এ ইনহেরিট্যান্স খুবই গুরুত্বপূর্ণ এবং কাজের একটা জিনিস। এটার সাহায্যে চাইলে সহজেই একটা অবজেক্ট এর ফিচার আরেকটা অবজেক্ট এ নেওয়া যায়। এখন এর মানে এখানে ক্লিয়ার না হলেও একটু বাস্তব উদাহরণ দিলে ঠিকমতো বুঝতে পারবেন।

ধরি আমাদের একটা ফাংশন কন্সট্রাক্টর আছে **Person** নামে। এখন একটা পার্সনের কি কি থাকতে পারে? নাম (**name**), বয়স(**age**), কি করে(**job**) এরকম কিছু থাকতে পারে। তাইলে এর জন্যে একটা ব্লু-প্রিন্ট তৈরী করলে এরকম হবেঃ

```
var Person = function(name, age, job) {  
  this.name = name;  
  this.age = age;  
  this.job = job;  
}
```

এখন এখান থেকে আমরা ইচ্ছামতো পার্সনের জন্যে অবজেক্ট তৈরী করতে পারবো। কিন্তু এখন যদি ধরি আমরা আরেকটা ব্লু-প্রিন্ট চাই **Teacher** নামে। এখন টিচারের কি কি থাকতে পারে? ধরে নিলাম টিচারেরও নাম(**name**), বয়স(**age**) ও কি করে(**job**) এগুলো আছে। সেই সাথে অতিরিক্ত আরেকটা আছে কোন সাবজেক্ট(**subject**) পড়ায়। এখন এই টিচারের জন্যে ব্লু-প্রিন্ট তৈরী করতে চাইলেঃ

```
var Teacher = function(name, age, job, subject) {  
  this.name = name;  
  this.age = age;  
  this.job = job;  
  this.subject = subject;  
}
```

কিন্তু এভাবে আমরা সেইম জিনিসগুলো পুনরায় ইউজ করাটা খুব ক্লিন দেখায় না। আমরা এখানে এমন কিছু করতে চাই যাতে আমরা **Person** থেকে পার্সনের ফিচারগুলো **Teacher** এ আনবো, এবং সেই সাথে টিচারের অতিরিক্ত যেটা আছে সেটা ডিফাইন করে দিবো। এটাকেই মূলত ইনহেরিট করা বলে। এখন ইনহেরিট্যান্স কয়েকরকমের হতে পারে। নিচে আমি বিস্তারিত আলোচনা করলাম।

## প্রোটোটাইপ বেসড ইনহেরিট্যান্সঃ

ধরি গত পর্বে আমরা একটা ফাংশন কন্সট্রাকটর তৈরী করেছিলাম এরকমঃ

```
var PersonOnly = function(name, age, job) {  
  this.name = name;  
  this.age = age;  
  this.job = job;  
}
```

ধরি আমরা এই কন্সট্রাকটরটা ডিফাইন করে ফেলছি অলরেডি। কিন্তু এখন জন্ম তারিখ() বের করার জন্যে আরেকটা মেথড ঢুকাতে চাই এটার ভিতরে। কি করা যায়? আবার নতুন করে ডিফাইন করবো?? না! অবশ্যই না। এখান আমরা **prototype** ইউজ করে নতুন মেথডটাকে আগের থাকা **PersonOnly** এ ঢুকাতে পারবো এভাবেঃ

```
PersonOnly.prototype.dateOfBirth = function() {  
  console.log(this.name + ' born in ' + (2018 -  
  this.age));  
}
```

এখন কন্সোলে এটা লিখুনঃ

```
console.log(PersonOnly.prototype);
```

দেখবেন এখানে আমাদের নতুন তৈরী করা মেথডটা আছেঃ

```
> console.log(PersonOnly.prototype);
▼ {dateOfBirth: f, constructor: f} 1
  ▶ dateOfBirth: f ()
  ▶ constructor: f (name, age, job)
  ▶ __proto__: Object
```

এখন আমরা প্রোটোটাইপ চেইনের কথা আগের পর্বে জেনেছি। প্রোটোটাইপ চেইনের কারণে এখন এই মেথডটাকেও অ্যাক্সেস করা যাবে এই **PersonOnly** থেকে তৈরী করা অবজেক্টগুলো থেকে। ধরি নিচে একটা অবজেক্ট তৈরী করলাম এই কন্সট্রাকটর ফাংশন থেকেঃ

```
var zawad = new PersonOnly('Zawad Ahmed', 1, 'Child');
```

এখন আমরা আমাদের তৈরীকৃত মেথডটাকে অ্যাক্সেস করতে চাইলেঃ

```
zawad.dateOfBirth();
```

```
> zawad.dateOfBirth();
Zawad Ahmed born in 2017
```

ব্যাস! এভাবেই আমরা অ্যাক্সেস করে ফেলতে পারতেছি। একইরকমভাবে আমরা ভ্যারিয়েবলকেও কন্সট্রাকটর ফাংশনে দিতে পারবো। যেমন আমরা নতুন আরেকটা ভ্যারিয়েবল দিলাম **PersonOnly** এঃ

```
PersonOnly.prototype.address = 'Bangladesh';
```

এখন এটা আমরা **zawad** অবজেক্ট থেকেও অ্যাক্সেস করতে পারবোঃ

```
console.log(zawad.address)
```

```
> console.log(zawad.address)
Bangladesh
```

ম্যাজিক! এটাই ইনহেরিট্যান্স এর ম্যাজিক। এবার আসুন আমরা আরো বড় কিছু দিকে যাই।

## কন্সট্রাকটর ইনহেরিট্যান্সঃ

যেটা প্রথমে আমরা আলোচনা করেছিলাম। একটা পার্সন আর একটা টিচারের মধ্যে একটা বৈশিষ্ট্য খালি পার্থক্য রয়েছে। এখন আমাদের সেজন্যে কি দুইটা পৃথক পৃথক ফাংশন কন্সট্রাকটর তৈরী করতে হবে? অবশ্যই না! আমাদের ইনহেরিট্যান্স আছে কি জন্যে?

ধরি আমাদের একটা ফাংশন কন্সট্রাকটর আছে **Person** নামেঃ

```
var Person = function(name, age, job) {
  this.name = name;
  this.age = age;
  this.job = job;
```

```
}
```

এখন আমাদের টিচারদের জন্যেও একটা ফাংশন কন্সট্রাকটর দরকার যেখানে ঠিক এইগুলোও থাকবে সেই সাথে অতিরিক্ত একটা **subject** নামে আরেকটা থাকবে। এখন এটার জন্যে ফাংশন কন্সট্রাকটর এমন হতে পারেঃ

```
var Teacher = function(name, age, job, subject) {  
  this.name = name;  
  this.age = age;  
  this.job = job;  
  this.subject = subject;  
}
```

এখন আমাদের ইনহেরিট্যান্স থাকার কারনে এভাবে নতুন করে **Teacher** এর ভিতরে সবকিছু ডিফাইন করার কোনো দরকার নাই। আমরা চাইলে সহজেই **Person** থেকে **name**, **age** আর **job** নিয়ে আসতে পারি এভাবেঃ

```
var Teacher = function(name, age, job, subject) {  
  Person.call(this, name, age, job);  
  this.subject = subject;  
}
```

নতুন নতুন কিছু দেখা যাচ্ছে মনে হয়? হ্যাঁ, এটা একটু বুঝার জিনিস। এখানে আমরা ভিতরে **Person** কে কল করে এনেছি এবং **call** মেথড ইউজ করে **this** এর ভ্যালু ডিটারমাইন করে দিলাম। এখন

এখানে `this` মানে `this` ই! কেনো? হ্যাঁ মনে করার চেষ্টা করুন। আমরা যখন এটা দিয়ে অবজেক্ট তৈরী করবো, তখন `new` কীওয়ার্ড ইউজ করবো। আর সেজন্যে আমাদের `this` এর নতুন কন্টেক্সট তৈরী হবে যেটা নতুন অবজেক্টটাকে ইন্ডিকেট করবে। তার মানে এখানে ডিফাইন করা `this` টা আসলে আমরা ভবিষ্যতে যে অবজেক্ট তৈরী করবো সেটাকেই ইন্ডিকেট করছে। এখন বাকি অংশ ঠিক আগের মতোইঃ

```
var kamaljeet = new Teacher('Kamaljeet Saini', 53,
  'Teacher', 'CSE');
```

ব্যাস এভাবেই আমাদের নতুন অবজেক্ট তৈরী করা যাবে। এটা ঠিকঠাকমতোই কাজ করবেঃ

```
console.log(kamaljeet.name);
console.log(kamaljeet.age);
console.log(kamaljeet.job);
console.log(kamaljeet.subject);
```

```
> console.log(kamaljeet.name);
  console.log(kamaljeet.age);
  console.log(kamaljeet.job);
  console.log(kamaljeet.subject);
Kamaljeet Saini
53
Teacher
CSE
```

ব্যাস! আমরা এভাবেই `Person` এর ফিচারগুলো `Teacher` এ ইনহেরিট করে নিয়ে আসলাম। এটা



কিছুটা হলেও আমাদের লাইফ সহজ করে দিবে।

এখন আমাদের এমনও ফাংশন কন্সট্রাকটরও থাকতে পারে যেটা কোনো প্যারামিটার নেয় না। বাই ডিফল্ট ভ্যালু সেট করা থাকতে পারে এরকমঃ

```
var GeneralPeople = function(name, age, job) {  
  this.name = name;  
  this.age = age;  
  this.job = job;  
}
```

এখন ধরি আরেকটা অবজেক্ট আছে **Athlete** নামে যেটা ঠিক **GeneralPeople** এর সবগুলো ফিচারই থাকবে। সাথে এক্সট্রা আরেকটা থাকবে। তো এখানেও সেইমভাবে ইনহেরিট্যান্স ইউজ করা যাবেঃ

```
var Athlete = function(name, age, job, prize) {  
  GeneralPeople.call(this);  
  this.prize = 100;  
}
```

ব্যাস এভাবেই আমরা কোনো প্যারামিটার/আর্গুমেন্ট ছাড়াও ইনহেরিট্যান্স নিয়ে কাজ করতে পারি।

এখানে দুইটার মধ্যে কোনো পার্থক্যই নেই আসলে।

আশা করি ইনহেরিট্যান্স নিয়ে আর কোনো সমস্যা থাকবে না। এখন থেকে আপনি রিয়েল লাইফ কোনো জিনিসেও ইনহেরিট্যান্স কাজে লাগাতে পারবেন আর আপনার ডেভেলপমেন্ট এর লাইফ সহজ করতে

পারবেন।

---

---

---

## 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

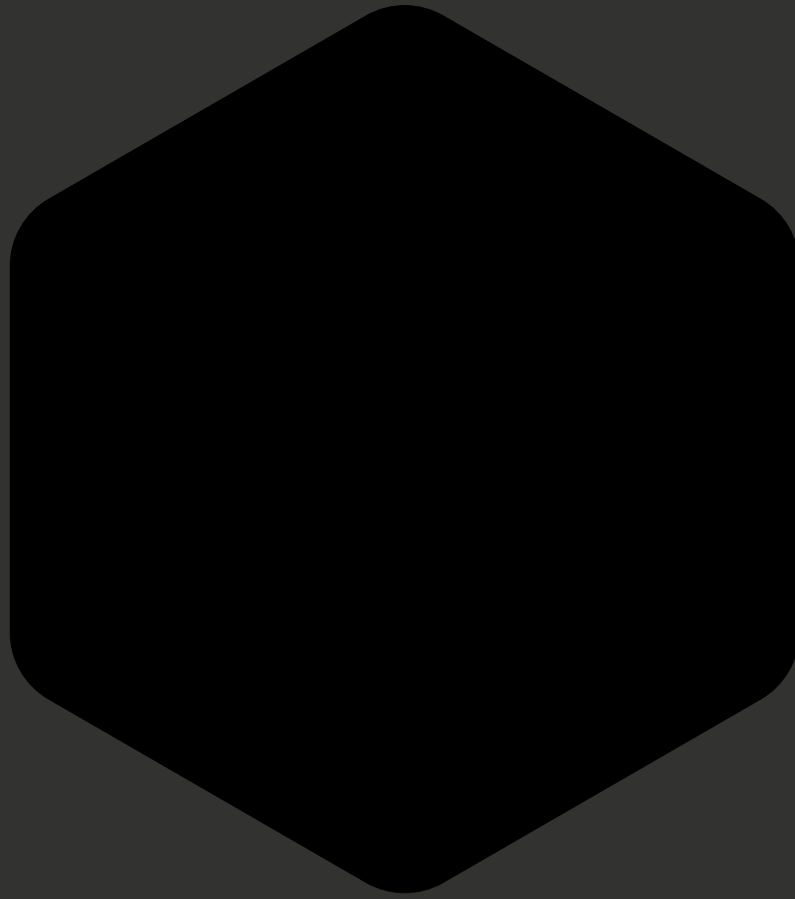
## রিসোর্সঃ

এমডিএন ডকুমেন্টেশন

ইউডেমি কোর্স

Eloquent JavaScript

You Don't Know JS



♥ এর সাথে ডেভেলপ করেছে **জুনায়েদ আহমেদ**