

Module 36-(advanced) JavaScript Object Concepts

ইন্টারভিউ প্রশ্ন

আজকের মডিউল একটু কঠিন কারণ লাষ্টের জিনিসগুলো ইন্টারভিউতে মাঝে মধ্যে জিজ্ঞেস করে বসে। তো আজকের মডিউল থেকে কি কি ইন্টারভিউ প্রশ্ন আসতে পারে। (এই প্রশ্নগুলোর উত্তর এখন না পারলেও সমস্যা নাই। পরে দেখে ফেলবে। ভিডিওতে তো আজীবন একসেস থাকবে)

১. (এডভান্সড) জাভাস্ক্রিপ্ট এ দুইটা অবজেক্ট সেইম কিনা সেটা কিভাবে কম্পায়ার করে
২. (এডভান্সড) bind, call, apply এর মধ্যে ডিফারেন্স কি
৩. (এডভান্সড) জাভাস্ক্রিপ্ট এর মধ্যে this কীওয়ার্ড কিভাবে কাজ করে
৪. arrow ফাংশন আর রেগুলার ফাংশন এর মধ্যে ডিফারেন্স কি

36-1 Module introduction Different way to Create Object

There are four ways to create an object in JavaScript - **using object literals**, **using the function constructor**, **using the Object.create method**, and **using the class keyword** (which is almost the same as using a function constructor).

// Different way to Create Object

// 1. using object literal

```
const student = { name: "Sakib AL Hasan", job: "cricketer" };
```

// 2. object constructor

```
const person=new Object();
```

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

// 3.

```
// const human=Object.create(null);
```

```
// console.log(human);
```

```
const human=Object.create(student);
```

```
// console.log(human.job);
```

```
class People{
  constructor(name,age)
  {
    this.name=name;
    this.age=age;
  }
}
```

```

}

// 4. class
const peop=new People('Hamid Hosen',23);
// console.log(peop);

// function=> syntactical Sugar
function Manus(names){
    this.names=names;
}

// 5. function
const man=new Manus('Hamid Hosen')
console.log(man);
console.log(man.names);
.

```

36-2 Object method property review

Accessing Object Properties: objectName.methodName()

```

const student = {
    id: 101,
    money: 5000,
    name: "RJ Kibria",
    major: "mathematics",
    isRich: false,
    subjects: ["english", "economics", "math 101", "calculus"],
    bestFriend: {
        name: "kundu",
        major: "mathematics",
    },
    takeExam: function () {
        console.log(this.name, "taking exam");
    },
    treatDey: function (expense, boksis) {
        this.money = this.money - expense - boksis;
        return this.money;
    },
};

student.takeExam();
const remaining1=student.treatDey(900,100);
console.log(remaining1);

```

```
const remaining2=student.treatDey(500,50);
console.log(remaining2);

console.log(student.name);
```

36-3 Keys, values, entries, delete, seal, freeze

Object.keys(objectName) : Object এর property র নামগুলো array হিসেবে return করবে এই method এর মাধ্যমে।

Object.values(objectName) : Object এর property র value গুলো array হিসেবে return করবে এই method এর মাধ্যমে।

Delete objName.property; Object এর property এবং property র value দুটোকেই delete করে।

Object.seal():

(No) : Delation of existing Property

(No) : Insertion of new Property

(Yes) : Modification of new property

Object.freeze()

(No): Delation of existing Property

(No): Inserion of new Property

(No): Modification of new Property

```
const bottle = {
  color: "yellow",
  hold: "water",
  price: 50,
  isCleaned: true,
};

// getting all properties names
const keys=Object.keys(bottle);
// console.log(keys);
```

```

// get all values
const values=Object.values(bottle);
// console.log(values);

const pairs=Object.entries(bottle);
// console.log(pairs);

// seal() use করলে delete করা যায় না।
// Object.seal(bottle);
Object.freeze(bottle);
bottle.price=100;
bottle.height=200;
delete bottle.isCleaned;
console.log(bottle);

```

36-4 Loop through an object using for in, for of, object entries

```

const bottle = {
  color: "yellow",
  hold: "water",
  price: 50,
  isCleaned: true,
};

/*
for(let i = 0; i<10;i++){
for(const num of numbers){} // array
for(const prop in student){} // object
*/

for (const prop in bottle) {
  // console.log(prop,bottle[prop]);
}

const keys = Object.keys(bottle);
// console.log(keys);
for (const prop of keys) {
  console.log(prop, bottle[prop]);
}

// advanced
const entries = Object.entries(bottle);
// console.log(entries);

```

```
// const [key, value] = ['color', 'yellow'];
for (const [key, value] of Object.entries(bottle)) {
  console.log(key, value);
}

//
// const arr = ['color', 'jeans', 'tom']
// arr['color']
```

36-5 (advanced) Compare objects, referential integrity

36-6 (advanced) Use bind to borrow method from another object

The bind() method creates a new function, when invoked, has the this sets to a provided value. **The bind() method allows an object to borrow a method from another object without making a copy of that method.** This is known as function borrowing in JavaScript.

36-7 (advanced) Difference between bind, call and apply

	Call()	Apply()	Bind()
Execution	At the time of binding.	At the time of binding.	At the time when we execute the return function
Parameter	Any number of arguments one by one.	Array[]	Array and any number of arguments.

(এডভান্সড) bind, call, apply এর মধ্যে ডিফারেন্স কি ;

36-8 (super advanced) Understand this keyword in JavaScript

- This হচ্ছে execution context.
- Regular method হলে বাম পাশে যে আছে সেটাকে context হবে।
- Arrow function হলে উপরের level অনুসারে context ধরবে।
- কোন dom এর element এ click করলে, সেই event টাই This বুঝবে।
- Dom এর element এর event টা কোন event handler হলে (যেটা পরে execute হবে), তাহলে function টা calling এর উপর This নির্ভর করবে।

36-9 Module summary and checklist