

closure in Javascript

```
function asynop()
{
  let c = 0;
  return function() {
    c += 1;
    console.log('opn called ${count}');
  };
}

const opn = asynop();
opn();
opn();
```

Here, each call to 'opn()' remembers the value of 'c' because of closure

Prototype in javascript

```
function User(name, age) {
  this.name = name;
  this.age = age;
}
```

// Adding a method to User prototype.

```
User.prototype.greet = function() {
  console.log('Hello, ${name}, ${age}');
};
```

```
const user1 = new User('Alice', 30);
const user2 = new User('Aman', 20);
```

Modules and Import/Export in Node.js

→ Exporting a Module:

```
function add(a, b) {
  return a + b;
}

module.exports = add;
```

→ Importing a Module:

```
const add = require('./calc');
console.log(add(2, 3));
```

→ Exporting in ES6:

```
export function add(a, b)
{
  return a + b;
}
```

→ Importing in ES6:

```
import { add } from './calc';
console.log(add(2, 3));
```


Error Handling Techniques in Node.js -

```

function func() {
  if (Math.random() < 0.5) {
    throw new Error('Something went wrong');
  }
  return 'Success';
}

try {
  const res = func();
  console.log(res);
} catch (error) {
  console.log(error.message);
}

```

→ Error Handling in Asynchronous code (Promises) -

```

function asyncfun() {
  return new Promise((resolve, reject) => {
    setTimeout(() => {
      const success = Math.random() > 0.5;
      if (success)
        resolve('opn successful');
      else
        reject(new Error('opn failed'));
    }, 1000);
  });
}

```


The power of Apply, Call & Bind in Node.js

```
⇒ function greet(){  
  console.log('Hello, I am ${this.name}');  
}
```

```
const user = {name: 'Alice'};  
greet.call(user);
```

```
⇒ function update(age, occ){  
  this.age = age;  
  this.occ = occ;  
}
```

```
const person = {name: 'Aman'};  
update.apply(person, [30, 'Developer']);  
console.log(person);
```

```
⇒ function log(){  
  console.log('Logged in as ${this.username}');
```

```
}  
const session = {username: 'user123'};
```

```
const boundLog = log.bind(session);  
boundLog();
```