

**JavaScript** একটি শক্তিশালী স্ক্রিপ্টিং ভাষা যা ওয়েব ডেভেলপমেন্টে ব্যাপকভাবে ব্যবহৃত হয়। এটি ডাইনামিক কনটেন্ট তৈরি করতে এবং ইউজার ইন্টারঅ্যাকশন প্রসেস করতে সাহায্য করে। এখানে JavaScript-এর কিছু বেসিক সিনট্যাক্স উপাদান সম্পর্কে বিস্তারিত আলোচনা করা হল: ভেরিয়েবল, লুপ, ফাংশন, এবং অবজেক্ট।

## ১. ভেরিয়েবল (Variables)

JavaScript-এ ভেরিয়েবল ডিক্লেয়ার করার জন্য তিনটি কীওয়ার্ড ব্যবহার করা হয়: `var`, `let`, এবং `const`।

- **var:** পুরনো স্টাইল, যা স্কোপ পরিবর্তনশীলতা নিয়ে কিছু সমস্যা তৈরি করতে পারে।
- **let:** ব্লক স্কোপে ভেরিয়েবল ডিক্লেয়ার করতে ব্যবহৃত হয়।
- **const:** কনস্ট্যান্ট ভেরিয়েবল ডিক্লেয়ার করতে ব্যবহৃত হয়, যার মান পরিবর্তন করা যায় না।

### উদাহরণ:

```
// var ব্যবহার
var name = "John";
name = "Doe";

// let ব্যবহার
let age = 30;
age = 31;

// const ব্যবহার
const birthYear = 1990;
// birthYear = 1991; // ত্রুটি হবে
```

## ২. লুপ (Loops)

JavaScript-এ বিভিন্ন ধরনের লুপ রয়েছে যা কোড ব্লক বারবার চালাতে ব্যবহৃত হয়।

### • for লুপ:

```
for (let i = 0; i < 5; i++) {
  console.log(i); // আউটপুট: 0, 1, 2, 3, 4
}
```

### • while লুপ:

```
let i = 0;
while (i < 5) {
  console.log(i); // আউটপুট: 0, 1, 2, 3, 4
  i++;
}
```

### • do...while লুপ:

```
let i = 0;
do {
  console.log(i); // আউটপুট: 0, 1, 2, 3, 4
  i++;
} while (i < 5);
```

## ৩. ফাংশন (Functions)

ফাংশন একটি ব্লক কোড যা একটি নাম দিয়ে ডিফাইন করা হয় এবং পরবর্তীতে কল করা যায়।

### • ফাংশন ডিফিনিশন:

```
function greet(name) {
```

```
    return "Hello, " + name;
}
console.log(greet("Alice")); // আউটপুট: Hello, Alice
```

- **ফাংশন এক্সপ্রেশন:**

```
const multiply = function(a, b) {
    return a * b;
};
console.log(multiply(2, 3)); // আউটপুট: 6
```

- **আরো শর্টকাট সিনট্যাক্স (Arrow Functions):**

```
const add = (a, b) => a + b;
console.log(add(5, 3)); // আউটপুট: 8
```

## 8. অবজেক্ট (Objects)

JavaScript-এ অবজেক্ট এক বা একাধিক প্রপার্টি এবং তাদের মান ধারণ করে। অবজেক্ট কীগুলির মাধ্যমে মান অ্যাক্সেস করা যায়।

- **অবজেক্ট ডিফিনিশন:**

```
let person = {
    name: "John",
    age: 30,
    greet: function() {
        return "Hello, " + this.name;
    }
};

console.log(person.name); // আউটপুট: John
console.log(person.greet()); // আউটপুট: Hello, John
```

- **অবজেক্ট প্রোপার্টি অ্যাক্সেস এবং মডিফিকেশন:**

```
// প্রপার্টি অ্যাক্সেস
console.log(person["age"]); // আউটপুট: 30

// প্রপার্টি আপডেট
person.age = 31;
console.log(person.age); // আউটপুট: 31
```

- **অবজেক্ট ক্রিয়েশন:**

```
let car = new Object();
car.make = "Toyota";
car.model = "Corolla";
car.year = 2020;

console.log(car.make); // আউটপুট: Toyota
```

## সারাংশ:

JavaScript-এ ভেরিয়েবল, লুপ, ফাংশন, এবং অবজেক্ট হল কিছু মৌলিক উপাদান যা কোড লেখার সময় প্রায়ই ব্যবহার করা হয়। ভেরিয়েবল বিভিন্ন ধরনের ডেটা স্টোর করতে সাহায্য করে, লুপ বিভিন্ন কার্যক্রম পুনরাবৃত্তি করতে ব্যবহৃত হয়, ফাংশন কোড ব্লককে পুনরায় ব্যবহারযোগ্য করে এবং অবজেক্ট জটিল ডেটা স্ট্রাকচার তৈরি করতে সাহায্য করে।

**JavaScript-এ DOM (Document Object Model)** ম্যানিপুলেশন ওয়েব পেজের কনটেন্ট এবং স্ট্রাকচার পরিবর্তন করতে ব্যবহৃত হয়। DOM হল একটি পেজের স্ট্রাকচারাল রিপ্রেজেন্টেশন যা HTML ডকুমেন্টকে ট্রী আকারে রেন্ডার করে। DOM ম্যানিপুলেশন দ্বারা আপনি HTML এলিমেন্ট অ্যাড, রিমুভ, অথবা আপডেট করতে পারেন, এবং ইভেন্ট হ্যান্ডলার অ্যাটাচ করতে পারেন।

## DOM ম্যানিপুলেশন বেসিক

### ১. DOM এলিমেন্ট সিলেকশন

#### getElementById

- একটি নির্দিষ্ট ID সহ এলিমেন্ট নির্বাচন করতে ব্যবহৃত হয়।

```
let element = document.getElementById('myElement');
```

#### getElementsByClassName

- নির্দিষ্ট ক্লাস নাম সহ সমস্ত এলিমেন্ট নির্বাচন করতে ব্যবহৃত হয়।

```
let elements = document.getElementsByClassName('myClass');
```

#### getElementsByTagName

- নির্দিষ্ট ট্যাগ নাম সহ সমস্ত এলিমেন্ট নির্বাচন করতে ব্যবহৃত হয়।

```
let elements = document.getElementsByTagName('div');
```

#### querySelector

- CSS সিলেক্টর ব্যবহার করে একটি এলিমেন্ট নির্বাচন করতে ব্যবহৃত হয়।

```
let element = document.querySelector('.myClass');
```

#### querySelectorAll

- CSS সিলেক্টর ব্যবহার করে সমস্ত মেলানো এলিমেন্ট নির্বাচন করতে ব্যবহৃত হয়।

```
let elements = document.querySelectorAll('div.myClass');
```

### ২. এলিমেন্টের কনটেন্ট পরিবর্তন

#### innerHTML

- এলিমেন্টের HTML কনটেন্ট পরিবর্তন করতে ব্যবহৃত হয়।

```
let element = document.getElementById('myElement');  
element.innerHTML = '<p>New Content</p>';
```

#### textContent

- এলিমেন্টের টেক্সট কনটেন্ট পরিবর্তন করতে ব্যবহৃত হয়।

```
let element = document.getElementById('myElement');  
element.textContent = 'New Text Content';
```

### ৩. এলিমেন্ট অ্যাট্রিবিউট পরিবর্তন

#### getAttribute এবং setAttribute

- এলিমেন্টের অ্যাট্রিবিউট পড়া এবং পরিবর্তন করতে ব্যবহৃত হয়।

```
let element = document.getElementById('myElement');
```

```
let src = element.getAttribute('src'); // অ্যাট্রিবিউট পড়া
element.setAttribute('src', 'newImage.jpg'); // অ্যাট্রিবিউট পরিবর্তন
```

## ৪. এলিমেন্ট স্টাইল পরিবর্তন

### style

- এলিমেন্টের inline CSS স্টাইল পরিবর্তন করতে ব্যবহৃত হয়।

```
let element = document.getElementById('myElement');
element.style.color = 'blue';
element.style.fontSize = '20px';
```

## ৫. এলিমেন্ট অ্যাড এবং রিমুভ

### appendChild এবং removeChild

- একটি নতুন এলিমেন্ট অ্যাড অথবা বিদ্যমান এলিমেন্ট রিমুভ করতে ব্যবহৃত হয়।

```
let parent = document.getElementById('parentElement');

// এলিমেন্ট তৈরি এবং অ্যাড
let newElement = document.createElement('div');
newElement.textContent = 'New Element';
parent.appendChild(newElement);

// এলিমেন্ট রিমুভ
let child = document.getElementById('childElement');
parent.removeChild(child);
```

## ৬. ইভেন্ট হ্যান্ডলার

### addEventListener

- ইভেন্ট হ্যান্ডলার অ্যাটাচ করতে ব্যবহৃত হয়।

```
let button = document.getElementById('myButton');
button.addEventListener('click', function() {
    alert('Button Clicked!');
});
```

### removeEventListener

- পূর্বে অ্যাটাচ করা ইভেন্ট হ্যান্ডলার রিমুভ করতে ব্যবহৃত হয়।

```
function handleClick() {
    alert('Button Clicked!');
}
button.addEventListener('click', handleClick);
button.removeEventListener('click', handleClick);
```

## সম্পূর্ণ উদাহরণ

### HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>DOM Manipulation</title>
</head>
<body>
    <div id="container">
        <p id="para1">This is a paragraph.</p>
        <button id="myButton">Click Me</button>
```

```
</div>

<script src="script.js"></script>
</body>
</html>
```

### JavaScript (script.js):

```
// এলিমেন্ট নির্বাচন
let para = document.getElementById('para1');
let button = document.getElementById('myButton');

// এলিমেন্টের কনটেন্ট পরিবর্তন
para.textContent = 'Updated paragraph text.';

// ইভেন্ট হ্যান্ডলার অ্যাটচ
button.addEventListener('click', function() {
  para.textContent = 'Paragraph text changed on button click.';
});

// নতুন এলিমেন্ট অ্যাড
let newElement = document.createElement('p');
newElement.textContent = 'New paragraph added.';
document.getElementById('container').appendChild(newElement);
```

## সারাংশ

JavaScript-এ DOM ম্যানিপুলেশন পেজের কনটেন্ট, স্ট্রাকচার, এবং স্টাইল পরিবর্তন করার ক্ষমতা প্রদান করে। এটি আপনাকে ডাইনামিক ইন্টারফেস তৈরি করতে এবং ব্যবহারকারীর ইন্টারঅ্যাকশনের উপর ভিত্তি করে পেজ আপডেট করতে সাহায্য করে।

**ES6 (ECMAScript 2015)** এবং তার পরের সংস্করণগুলির নতুন ফিচারগুলি JavaScript-কে আরও শক্তিশালী ও ব্যবহারযোগ্য করে তুলেছে। এখানে কিছু গুরুত্বপূর্ণ ফিচার এবং তাদের উদাহরণ দেওয়া হলো:

## ১. let এবং const

**let এবং const** নতুন ভেরিয়েবল ডিক্লেয়ার করার জন্য ব্যবহৃত হয় এবং এগুলি var এর তুলনায় ব্লক স্কোপের সাথে যুক্ত।

- **let:** একটি ভেরিয়েবল ব্লক স্কোপের মধ্যে ডিক্লেয়ার করে, যা var এর তুলনায় আরো সঠিক।

```
// let উদাহরণ
function example() {
  if (true) {
    let x = 10;
    console.log(x); // আউটপুট: 10
  }
  console.log(x); // ReferenceError: x is not defined
}
example();
```

- **const:** একটি কনস্ট্যান্ট ভেরিয়েবল ডিক্লেয়ার করে যার মান পরিবর্তন করা যায় না। এটি ব্লক স্কোপড।

```
// const উদাহরণ
const y = 20;
console.log(y); // আউটপুট: 20
```

```
y = 30; // TypeError: Assignment to constant variable.
```

## ২. অ্যারো ফাংশন (Arrow Functions)

**অ্যারো ফাংশন** ES6 এ একটি নতুন সিনট্যাক্স যা ফাংশন ডিফাইন করার জন্য সহজ ও সংক্ষিপ্ত উপায় প্রদান করে। এগুলি সাধারণত আনোনিমাস ফাংশনের জন্য ব্যবহৃত হয়।

- **সাধারণ ফাংশন:**

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

- **অ্যারো ফাংশন:**

```
const add = (a, b) => a + b;
```

**একই ভাবে, যদি একটি আর্গুমেন্ট থাকে, প্যারেন্থেসিস ব্যবহার করার দরকার নেই:**

```
const square = x => x * x;
```

**ব্লক স্টেটমেন্ট সহ অ্যারো ফাংশন:**

```
const multiply = (a, b) => {  
  const result = a * b;  
  return result;  
};
```

## ৩. মডিউল (Modules)

**মডিউল** ES6 এ একটি নতুন ফিচার যা কোডের সংগঠন এবং পুনঃব্যবহারযোগ্যতা উন্নত করতে ব্যবহৃত হয়। মডিউলগুলি `import` এবং `export` কিওয়ার্ড ব্যবহার করে কোড ভাগ করতে সাহায্য করে।

- **মডিউল এক্সপোর্ট করা:**

**math.js**

```
// এক্সপোর্ট করা  
export const add = (a, b) => a + b;  
export const subtract = (a, b) => a - b;
```

- **মডিউল ইমপোর্ট করা:**

**main.js**

```
import { add, subtract } from './math.js';  
  
console.log(add(5, 3)); // আউটপুট: 8  
console.log(subtract(5, 3)); // আউটপুট: 2
```

**ডিফল্ট এক্সপোর্ট:**

**math.js**

```
const multiply = (a, b) => a * b;  
export default multiply;
```

**main.js**

```
import multiply from './math.js';  
  
console.log(multiply(5, 3)); // আউটপুট: 15
```

## 8. অন্যান্য নতুন ফিচার

- ডিফল্ট প্যারামিটারস:

```
function greet(name = 'Guest') {  
  console.log(`Hello, ${name}!`);  
}  
greet(); // আউটপুট: Hello, Guest!  
greet('Alice'); // আউটপুট: Hello, Alice!
```

- স্প্রেড অপারেটর (...):

- অ্যাারেতে স্প্রেড অপারেটর:

```
const arr1 = [1, 2, 3];  
const arr2 = [...arr1, 4, 5];  
console.log(arr2); // আউটপুট: [1, 2, 3, 4, 5]
```

- অবজেক্টে স্প্রেড অপারেটর:

```
const obj1 = { a: 1, b: 2 };  
const obj2 = { ...obj1, c: 3 };  
console.log(obj2); // আউটপুট: { a: 1, b: 2, c: 3 }
```

- ডেস্ট্রাকচারিং:

- অ্যাারেতে ডেস্ট্রাকচারিং:

```
const [x, y] = [1, 2];  
console.log(x); // আউটপুট: 1  
console.log(y); // আউটপুট: 2
```

- অবজেক্টে ডেস্ট্রাকচারিং:

```
const person = { name: 'John', age: 30 };  
const { name, age } = person;  
console.log(name); // আউটপুট: John  
console.log(age); // আউটপুট: 30
```

## সারাংশ

- **let এবং const:** ব্লক স্কোপ ভেরিয়েবল ডিক্লেয়ার করতে ব্যবহৃত হয়।
- **অ্যারো ফাংশন:** সংক্ষিপ্ত সিনট্যাক্স ফাংশন ডিফাইন করার জন্য।
- **মডিউল:** কোডকে ভাগ করে এবং পুনঃব্যবহারযোগ্য করে তোলা।
- **অন্যান্য ফিচার:** ডিফল্ট প্যারামিটারস, স্প্রেড অপারেটর, এবং ডেস্ট্রাকচারিং।

এই নতুন ফিচারগুলি JavaScript-কে আরো শক্তিশালী ও সহজবোধ্য করে তোলে, কোড লেখার সময় উন্নত সুবিধা প্রদান করে।