



# JavaScript Programming Master Notes

Created by: Mahedi Hasan Topic: String & Object Fundamentals

Email: mdmahedi.mpi@gmail.com

---

## ভিডিও ১: স্ট্রিং এবং অ্যারের প্রাথমিক ধারণা

JavaScript

```
const country = 'Bangladesh'
```

```
const division = "Newkalehe"
```

```
const district = `B-Baria`
```

```
console.log(typeof division) // output: string
```

```
const numbers = [35, 56, 98, 566, 785]
```

```
console.log(numbers.length) // output: 5
```

```
console.log(numbers[2]) // output: 98
```

/\*\*

\*  নোট:

\* অ্যারে (Array) হচ্ছে মিউটেবল (Mutable), অর্থাৎ এর ভেতরের উপাদান পরিবর্তন করা যায়।

\* কিন্তু স্ট্রিং (String) হলো ইম্মিউটেবল (Immutable), অর্থাৎ একবার তৈরি করলে এর কোনো নির্দিষ্ট ইনডেক্সের ভ্যালু সরাসরি পরিবর্তন করা যায় না।

\*/

```
const capital = 'Dhaka'
```

```
console.log(capital)
```

```
capital[3] = 'F' // এটি কাজ করবে না কারণ স্ট্রিং ইম্মিউটেবল
```

```
console.log(capital) // output: Dhaka
```

## ভিডিও ২: স্ট্রিং মেথডস (Case & Trim)

JavaScript

```
const school = 'Rajuk utara Model school'
```

```
console.log(school.toLowerCase()) // সব ছোট হাতের অক্ষর হবে
```

```
console.log(school.toUpperCase()) // সব বড় হাতের অক্ষর হবে
```

/\*\*

\*  নোট:

\* দুটি স্ট্রিং তুলনা করার সময় (Comparison) সবসময় সেগুলোকে একই কেসে (যেমন .toLowerCase())

\* রূপান্তর করে নেওয়া ভালো, নয়তো জাভাস্ক্রিপ্ট Case-sensitive হওয়ার কারণে রেজাল্ট ভুল আসতে পারে।

```
*/
```

```
const subjects = 'Chemistry'
```

```
const book = 'chemistry'
```

```
if (subjects.toLowerCase() === book.toLowerCase()) {  
  console.log('iam reading book aibar porikkhai fatau felmu')  
} else {  
  console.log('hudai hudai prista ultai ar vat khawar jonno')  
}
```

```
/**
```

```
 *  নোট:
```

```
 * স্ট্রিংয়ের শুরুতে বা শেষে যদি বাড়তি হোয়াইট স্পেস (Space) থাকে, তবে সেগুলো সরানোর জন্য .trim() ব্যবহার করা হয়।
```

```
 * এটি তুলনা বা কন্ডিশন চেক করার সময় নির্ভুল ফলাফল দেয়।
```

```
*/
```

```
const drink = 'water'
```

```
const liquid = ' water ' // এখানে স্পেস আছে
```

```
if (drink.trim() === liquid.trim()) {  
  console.log('pani ar opor nam life')  
} else {  
  console.log('somudre pani khaite pari na')  
}
```

### ভিডিও ৩: স্লাইস, স্প্লিট এবং জয়েন (String Operations)

JavaScript

```
/**
```

```
 *  নোট:
```

```
 * .slice() মেথডটি স্ট্রিংয়ের নির্দিষ্ট অংশ কেটে নিতে ব্যবহার করা হয়।
```

```
 * এর প্রথম প্যারামিটার হচ্ছে শুরুর ইনডেক্স এবং দ্বিতীয়টি হচ্ছে শেষের আগের ইনডেক্স।
```

```
*/
```

```
const address = 'andorkilla'
```

```
const part = address.slice(2, 5)
```

```
console.log(part); // output: dor
```

```
/**
```

```
 *  নোট:
```

```
 * .split() ব্যবহার করা হয় একটি স্ট্রিংকে ভাগ করে অ্যারেতে রূপান্তর করার জন্য।
```

```
 * প্যারামিটার হিসেবে যা দেওয়া হবে, স্ট্রিংটি ঠিক সেই জায়গা থেকে আলাদা হয়ে যাবে।
```

```
*/
```

```
const sentence = 'I am a good and hardworking person'
```

```
console.log(sentence.split('a'))
```

```
const friendsStr = 'Rahim, kahim, fahim, lahim, shamim'  
const friends = friendsStr.split(',')  
console.log(friends)
```

```
/**
```

```
 *  নোট:
```

```
 * অ্যারের উপাদানগুলোকে একটি স্ট্রিংয়ে যুক্ত করার জন্য .join() ব্যবহার করা হয়।
```

```
 * ব্রাকেটের ভেতর আপনি যা দিবেন (যেমন: '-', '|') তা দিয়েই এলিমেন্টগুলো যুক্ত হবে।
```

```
 */
```

```
const realfriend = ['Rahim', 'kahim', 'fahim', 'lahim', 'shamim']  
console.log(realfriend.join())  
console.log(realfriend.join('|'))  
console.log(realfriend.join('-'))
```

```
/**
```

```
 *  নোট:
```

```
 * দুটি বা তার বেশি স্ট্রিংকে একত্রে জোড়া দেওয়ার জন্য .concat() ব্যবহার করা হয়।
```

```
 * এছাড়া .includes() দিয়ে চেক করা যায় কোনো স্ট্রিংয়ের ভেতরে নির্দিষ্ট কোনো অক্ষর বা শব্দ আছে কি না।
```

```
 */
```

```
const frist = 'Mahedi'  
const last = 'Hasan'  
const fullname2 = frist.concat(' ').concat(last)  
console.log(fullname2)  
console.log(last.includes('x')) // output: false
```


---

## ভিডিও 8: স্ট্রিং রিভার্স করার নিয়ম (String Reverse)

JavaScript

```
const sentence = 'I am learning Web Dev.'
```

```
/**
```

```
 *  ব্যাখ্যা (For-of loop):
```

```
 * এখানে প্রতিটি 'letter' লুপের মাধ্যমে আলাদা করা হচ্ছে।
```

```
 * লজিকটি হলো: 'reverse = letter + reverse'। অর্থাৎ নতুন প্রতিটি অক্ষর আগের জন্মানো স্ট্রিংয়ের সামনে বসছে।
```

```
 * উদাহরণ: 'I' আসলো, তারপর ' ' আসলো (হয়ে গেল ' I'), তারপর 'a' আসলো (হয়ে গেল 'a I')। এভাবে উল্টে যাচ্ছে।
```

```
 */
```

```
let reverse = ""  
for (const letter of sentence) {  
  reverse = letter + reverse  
}  
console.log(reverse)
```

```

/**
 * 📝 ব্যাখ্যা (Standard For loop):
 * এখানে লুপটি ০ থেকে শুরু হয়ে স্ট্রিংয়ের শেষ পর্যন্ত চলেছে।
 * আগের মতোই প্রতিটি নতুন অক্ষরকে জমানো অক্ষরের বাম পাশে বসানো হচ্ছে, ফলে পুরো বাক্যটি উল্টে যাচ্ছে।
 */
let rev = "";
for (let i = 0; i < sentence.length; i++) {
  const letter = sentence[i]
  rev = letter + rev
}
console.log(rev)

/**
 * 📝 ব্যাখ্যা (Shortcut Method):
 * এটি সবচেয়ে সহজ উপায়। প্রথমে .split("") দিয়ে স্ট্রিংটিকে অ্যারে বানানো হয়,
 * তারপর অ্যারের .reverse() মেথড দিয়ে উল্টানো হয়, সবশেষে .join("") দিয়ে আবার স্ট্রিং করা হয়।
 */
const reversed = sentence.split("").reverse().join("")
console.log(reversed)

```

---

## 👤 ভিডিও ৫ : অবজেক্ট ডিক্লেয়ারেশন এবং প্রপার্টি এক্সেস

### JavaScript

```

/**
 * 📝 নোট (Object Declaration):
 * অবজেক্ট হলো একটি নন-প্রিমিটিভ ডাটা টাইপ যেখানে Key: Value জোড়ায় তথ্য রাখা হয়।
 * এটি অনেকটা বাস্তব জীবনের কোনো জিনিসের বৈশিষ্ট্যের মতো।
 */


const bottle = {
  brand: 'apple',
  price: 45,
  color: 'white',
  isClean: true
}

const person = {
  name: 'Sadar Uddin',
  age: 25,
  profession: 'developer',
  salary: 25000,
  married: true,
}

```

```
'fav places': ['bandarbon', 'santmartin', 'kukata']
}
```

```
/**
```

\*  নোট (Accessing Property):

\* অবজেক্টের প্রপার্টি দুইভাবে এক্সেস করা যায়:

\* ১. ডট নোটেশন (person.name) - যখন কী-র নাম সরাসরি জানা থাকে।

\* ২. ব্র্যাকেট নোটেশন (person['age']) - যখন কী-র নামে স্পেস থাকে বা ভেরিফেবল দিয়ে এক্সেস করতে হয়।

```
*/
```

```
console.log(person.proffesion) // Dot notation
```

```
console.log(person['fav places']) // Bracket notation
```

```
// ভ্যালু আপডেট করা
```

```
person.salary = 30000;
```

```
person['age'] = 26;
```

---

## ভিডিও 7 : অবজেক্ট অপারেশন এবং নেস্টেড অবজেক্ট

JavaScript

```
const computer = {
  brand: 'lebovo',
  price: 35000,
  pricessor: 'intel',
  hdd: '512gb',
  monitor: 'hp'
}
```

```
/**
```

\*  নোট:

\* Object.keys(obj) দিলে শুধু সব Key বা প্রপার্টি নামগুলোর অ্যারে পাওয়া যায়।

\* Object.values(obj) দিলে শুধু সব Value গুলোর অ্যারে পাওয়া যায়।

```
*/
```

```
const keys = Object.keys(computer);
```

```
const values = Object.values(computer);
```

```
/**
```

\*  নোট (Nested Object & Delete):

\* অবজেক্টের ভেতরে আরও অবজেক্ট বা অ্যারে থাকতে পারে (Nested)।

\* ডট দিয়ে ভেতরে ঢুকে ভ্যালু চেক করা যায়। আর 'delete' কিউয়ার্ড দিয়ে কোনো প্রপার্টি মুছে ফেলা যায়।

```
*/
```

```
const college = {
  name: 'vnc',
  class: [10, 11, 34],
```

```
    unique: {
      color: 'blue',
      result: { gap: 5, merit: 'Top' }
    }
  }
delete college.class;
```

---

## ভিডিও 8 : অবজেক্ট লুপিং এবং ডিক্লেয়ারেশন সিস্টেম

JavaScript

```
const mobile = {
  brand: 'samsung',
  price: '25000',
  color: 'black',
  camera: '12mp',
  isNew: true
}
```

/\*\*

\*  নোট:

\* অ্যারের জন্য আমরা 'for...of' লুপ ব্যবহার করি, কিন্তু অবজেক্টের জন্য 'for...in' লুপ ব্যবহার করতে হয়।

\*/

```
for (const prop in mobile) {
  console.log(prop, ': ', mobile[prop])
}
```

/\*\*

\*  নোট (Object Creation Systems):

\* অবজেক্ট বিভিন্নভাবে তৈরি করা যায়:

\* ১. অবজেক্ট লিটারেল: { } ব্যবহার করে।

\* ২. New Object: new Object() কনস্ট্রাক্টর ব্যবহার করে।

\* ৩. Object.create: অন্য অবজেক্টকে প্রোটোটাইপ হিসেবে ব্যবহার করে।

\*/

```
const pen = { brand: 'econo', price: 10, color: "black" }
const pencil = new Object()
const rubber = Object.create({ price: 210, color: 'red' })
```

# জাভাস্ক্রিপ্ট স্ট্রিং ও অবজেক্ট নোটবুক ভিডিওর সারসংক্ষেপ একনজরে

#### #### ১. স্ট্রিং এবং অ্যারের পার্থক্য

\* \*\*অ্যারে (Array):\*\* মিউটেবল বা পরিবর্তনযোগ্য।

\* \*\*স্ট্রিং (String):\*\* ইমিউটেবল বা অপরিবর্তনযোগ্য। ইনডেক্স ধরে স্ট্রিংয়ের অক্ষর বদলানো যায় না।

#### #### ২. গুরুত্বপূর্ণ স্ট্রিং মেথডস

\* `.toLowerCase()`: সব অক্ষর ছোট হাতের করা।

\* `.toUpperCase()`: সব অক্ষর বড় হাতের করা।

\* `.trim()`: স্ট্রিংয়ের দুপাশের বাড়তি স্পেস মুছে ফেলা।

\* `.slice(start, end)`: স্ট্রিংয়ের নির্দিষ্ট অংশ কেটে নেওয়া।

\* `.split()`: স্ট্রিংকে ভেঙে অ্যারে বানানো।

\* `.includes()`: স্ট্রিংয়ের ভেতরে কোনো কিছু আছে কি না তা চেক করা।

#### #### ৩. স্ট্রিং রিভার্স (উল্টানো)

\* লজিক: `reverse = letter + reverse` (লুপের মাধ্যমে)।

\* শর্টকাট: `str.split("").reverse().join("")`

#### #### ৪. অবজেক্ট (Object) হ্যান্ডলিং

\* \*\*এক্সেস করার নিয়ম:\*\* - ডট নোটেশন: `person.name`

- ব্র্যাকেট নোটেশন: `person['age']` (স্পেস বা ভেরিয়েবল থাকলে এটি জরুরি)।

\* \*\*লুপিং:\*\* অবজেক্টের জন্য `for...in` লুপ ব্যবহার করতে হয়।

\* \*\*কী ও ভ্যালু:\*\* `Object.keys(obj)` দিলে সব কী পাওয়া যায় এবং `Object.values(obj)` দিলে সব ভ্যালু পাওয়া যায়।

**"Created by Mahedi Hasan"**