

১. জাভাস্ক্রিপ্ট কি জিনিস এইটা কি জানো?

জাভাস্ক্রিপ্ট একটি স্ক্রিপ্টিং বা প্রোগ্রামিং ভাষা যা আপনাকে ওয়েব পৃষ্ঠাগুলিতে জটিল বৈশিষ্ট্যগুলি প্রয়োগ করতে দেয়।

উদাহরণ :

সময়মত কন্টেন্ট আপডেট, ইন্টারেক্টিভ ম্যাপ, অ্যানিমেটেড 2D/3D গ্রাফিক্স, স্ক্রলিং ভিডিও ইত্যাদি প্রদর্শন করা।

জাভাস্ক্রিপ্ট 1995 সালে ব্রেন্ডন আইক দ্বারা উদ্ভাবিত হয়েছিল। এটি নেটস্কেপ 2-এর জন্য তৈরি করা হয়েছিল এবং 1997 সালে ECMA-262 স্ট্যান্ডার্ডে পরিণত হয়েছিল।

২. জাভাস্ক্রিপ্ট কিভাবে কাজ করে সেটা কি জানো?

জাভাস্ক্রিপ্ট একটি interpreted language. এটি compiled language নয় যেমনটা c,c++,java। Environment or browser এ একটি ইঞ্জিন থাকা দরকার, যা মানব-পঠনযোগ্য সিনট্যাক্সে লেখা JS কোড নেয় এবং এটিকে মেশিন কোডে পরিণত করে। Chrome এর ক্ষেত্রে এই ইঞ্জিনটির নাম V8, firefox এ SpiderMonkey ইত্যাদি।

৩. ভেরিয়েবল কি জিনিস?

ভেরিয়েবল মানে যা পরিবর্তিত হতে পারে। জাভাস্ক্রিপ্টে, একটি ভেরিয়েবল ডেটা মান সংরক্ষণ করে যা পরে পরিবর্তন করা যেতে পারে। জাভাস্ক্রিপ্টে একটি ভেরিয়েবল ঘোষণা করতে সংরক্ষিত কীওয়ার্ড var ব্যবহার করতে হয়।

উদাহরণ:- var favouriteFootballer = "Cristiano Ronaldo";

৪. ভেরিয়েবল কিভাবে ডিক্লেয়ার করে ?

পাঁচটি কাজ করে একটি ভেরিয়েবল ডিক্লেয়ার করা যায়।

- প্রথমে var keyword,
- Meaningful ভেরিয়েবল নাম,
- (=) sign দিতে হবে,
- মান সংরক্ষণ,
- (;) sign দিতে হবে (optional)।

উদাহরণ:- var holiday = "Friday";

৫. ভেরিয়েবল এর মান কিভাবে চেঞ্জ করে বা আপডেট করে?

২টা variation আছে ভেরিয়েবল পরিবর্তন করার।

no1-

প্রথমে ভেরিয়েবল ডিক্লেয়ার করতে হবে. তারপর চেইঞ্জ করতে বাকিসব ঠিক রেখে শুধু মানটা পরিবর্তন করতে হয়।

যেমন:

ডিক্লেয়ার:

```
var holiday = "Friday";
```

চেইঞ্জ:

```
var holiday = "Sunday";
```

এমনটা শুধু var keyword এর ক্ষেত্রে হবে। এটি Recommended না।

no২-

প্রথমে ভেরিয়েবল ডিক্লেয়ার করতে হবে. তারপর চেইঞ্জ করতে ভার keyword বাদ দিয়ে বাকি সব ঠিক রেখে শুধু মানটা পরিবর্তন করতে হয়। যেমন:

ডিক্লেয়ার:

```
var holiday = "Friday";
```

চেইঞ্জ:

```
holiday = "Sunday";
```

এটি Recommended।

৬. কি কি ধরনের ভেরিয়েবল হয়। সেগুলো কি কি?

জাভাস্ক্রিপ্ট আপনাকে তিনটি প্রিমিটিভ ডেটা প্রকারের সাথে কাজ করতে দেয়: সংখ্যা, স্ট্রিং, এবং বুলিয়ান। জাভাস্ক্রিপ্ট আরও দুটি ডেটা প্রকারকেও সংজ্ঞায়িত করে, নাল এবং আন্ডিফাইন্ড।

- সংখ্যা যেমন ১,৩২,৫৪।
- স্ট্রিং যেমন "রহিম", "কারিম" ইত্যাদি।
- বুলিয়ান যেমন true, false।
- নাল বলতে কোনো মান নেই। (একটু confusing)
- আন্ডিফাইন্ড বলতে মান আছে কিন্তু তা জানা নাই। (একটু confusing)

৭. জাভাস্ক্রিপ্ট এ **primitive and non-primitive data types** কি কি ? উদাহরণ হিসেবে বলো।

সুতরাং primitive এবং non-primitive মধ্যে মৌলিক পার্থক্য হল যে:

১- primitive মানগুলি অপরিবর্তনীয় এবং non-primitive মানগুলি পরিবর্তনযোগ্য।

২- primitive মান দ্বারা সংরক্ষণ করা হয় non-primitive রেফারেন্স দ্বারা সংরক্ষণ করা হয়।

৩-জাভাস্ক্রিপ্টের primitive ডেটা প্রকারের মধ্যে রয়েছে সংখ্যা, স্ট্রিং, বুলিয়ান, অনির্ধারিত, নাল। নন-প্রিমিটিভ ডেটা টাইপের মধ্যে রয়েছে অ্যারে, অবজেক্ট।

৮. ভেরিয়েবল এর নাম কিভাবে কিভাবে ডিক্লেয়ার করতে হয়। কোন কোন জিনিস নাম এ লেখা যাবে না। অর্থাৎ **Variable** এর **naming convention** সম্পর্কে বলো।

ভেরিয়েবল এর নাম কিভাবে কিভাবে ডিক্লেয়ার করতে হয়। এর উত্তর ৪ নং প্রশ্নে দেয়া হয়েছে।

কোন কোন জিনিস নাম এ লেখা যাবে না ?

- শুরুতে নাম্বার (\$ ও _ বাদে) অন্য কোনো character ব্যবহার করা যাবে না।
- স্ট্রিং ব্যবহার করা যাবে না।
- কোনো Reserved keyword ব্যবহার করা যাবে না।
- ক্যামেলকেস ব্যবহার করা ভাল।
- Meaningful নাম দেয়া ভাল।

৯. দুইটা ভেরিয়েবল এর মধ্যে =, -, *, /, % কিভাবে করে?

(=) এইটা আসাইনমেন্ট অপারেটর(একটির মান আরএকটিতে বসানো), (-) বিয়োগ অপারেটর, (*) গুন অপারেটর, (/) ভাগ অপারেটর, (%) মডুলাস অপারেটর।

উদাহরণ:

```
var firstNumber = 12;  
var secondNumber = 4;
```

আসাইনমেন্ট অপারেটর

```
firstNumber = secondNumber;  
console.log(firstNumber); // output : 4
```

বিয়োগ অপারেটর:

```
console.log(firstNumber - secondNumber); // output : 8
```

গুন অপারেটর:

```
console.log(firstNumber * secondNumber); // output : 48
```

ভাগ অপারেটর:

```
console.log(firstNumber / secondNumber); // output : 3
```

মডুলাস অপারেটর:

```
console.log(firstNumber % secondNumber); // output : 0
```

১০. শর্টহ্যান্ড গুলো জানতে হবে। বিশেষ করে +=, -=, *=, /= জানতে হবে।

- $(+=)$ সংযোজন অ্যাসাইনমেন্ট অপারেটর একটি ভেরিয়েবলের সাথে সঠিক অপারেন্ডের মান যোগ করে এবং ভেরিয়েবলে ফলাফল বরাদ্দ করে।
- $(-=-)$ বিয়োগ অ্যাসাইনমেন্ট অপারেটর একটি ভেরিয়েবল থেকে ডান অপারেন্ডের মান বিয়োগ করে এবং ভেরিয়েবলে ফলাফল বরাদ্দ করে।
- $(*=)$ গুন অ্যাসাইনমেন্ট অপারেটর একটি ভেরিয়েবল থেকে ডান অপারেন্ডের মান গুন করে এবং ভেরিয়েবলে ফলাফল বরাদ্দ করে।
- $(/=)$ ভাগ অ্যাসাইনমেন্ট অপারেটর একটি ভেরিয়েবল থেকে ডান অপারেন্ডের মান ভাগ করে এবং ভেরিয়েবলে ফলাফল বরাদ্দ করে।

১১.. ++ এবং -- এর কাজ কি ? এইটা কি জানো।

++ হল ইনক্রিমেন্ট অপারেটর। এটি 1 পরিবর্তনশীল বৃদ্ধি। $x++$; $x = x + 1$ এর সমতুল্য; অথবা $x += 1$ এর সমতুল্য;

-- হল ডিক্রিমেন্ট অপারেটর। এটি 1 পরিবর্তনশীল হ্রাস। $x--$; $x = x - 1$ এর সমতুল্য; অথবা $x -= 1$ এর সমতুল্য;

১২ parseInt, parseFloat, toFixed এইগুলো কি করে?

parseInt ফাংশনটি একটি স্ট্রিং হিসাবে একটি মান গ্রহণ করে এবং পূর্ণসংখ্যা প্রদান করে।

parseFloat() ফাংশনটি একটি স্ট্রিং গ্রহণ করতে এবং এটিকে একটি ফ্লোটিং-পয়েন্ট নম্বরে রূপান্তর করতে ব্যবহৃত হয়।

toFixed() ফাংশনটি একটি সংখ্যাকে একটি স্ট্রিংয়ে রূপান্তর করে এবং এটি স্ট্রিংটিকে নির্দিষ্ট সংখ্যক দশমিকে রাউন্ড করে।

১৩. Array কি জিনিস। এইটা কিভাবে কাজ করে? কিভাবে Array ডিক্লেয়ার করতে হয়?

Array একটি non-primitive ডাটাইপ। জাভাস্ক্রিপ্ট অ্যারেগুলি ০ ইনডেক্সড: একটি অ্যারের প্রথম উপাদানটির ইনডেক্স 0 এ, দ্বিতীয়টি ইনডেক্স 1 এ, ত্রিতীয় ইনডেক্স ২ এভাবে চলবে।

Array ডিক্লেয়ার:

- ভার keyword লেখতে হবে।
- অ্যারে এর নাম দিতে হবে।
- $(=)$ sign দিতে হবে।
- তারপর $[]$ sign দিতে হবে এবং data গুলকে পাশাপাশি কমা দিয়ে separate রাখতে হবে।।
- $(;)$ sign দিতে হবে (optional)।

১৪. **array** এর মধ্যে কয়টা উপাদান (**element**) আছে সেটা কিভাবে বের করে।

`array.length` এই method টি ব্যবহার করতে হবে।

উদাহরণ:

```
var friends = ['রহিম', 'কারিম', 'জদু', 'কদু'];  
console.log(friends.length) // output-4
```

১৫. **array** এর উপাদান গুলো এর পজিশন (**index**) কিভাবে কাজ করে। কত দিয়ে শুরু হয়। এবং মান কিভাবে চেক হয়।

জ্যাক্সি-প্ট অ্যারেগুলি ০ ইনডেক্সড: একটি অ্যারের প্রথম উপাদানটির ইনডেক্স 0 এ, দ্বিতীয়টি ইনডেক্স 1 এ, ত্রিতিও ইনডেক্স ২ এভাবে চলবে।

১৬. কোন একটা উপাদানের **index** এর মান -১ বলতে কি বুঝায় ?

কোন একটা উপাদানের **index** এর মান -১ বলতে কি বুঝায় ওই উপাদানটি **array** তে নাই।

১৭. কিভাবে **index** দিয়ে কোন একটা **array** এর মধ্যে উপাদান এর মান খুঁজে বের করতে পারো

অ্যারে এর উপাদান গুলো **access** করতে আমরা অ্যারে এর ইন্ডেক্স ব্যবহার করব। এভাবে অ্যারে এর উপাদান অ্যাক্সেস করতে হয়।

`arrayName[index of which value you wanna get]`

উদাহরণ:-

```
var friends = ['রহিম', 'কারিম', 'জদু', 'কদু'];  
এই অ্যারে টার ২ নং ইন্ডেক্স এর ভ্যালু হবে
```

```
var valueOfIndexOne = friends[1];  
console.log(valueOfIndexOne);  
output:- 'কারিম'
```

১৮. কিভাবে কোন একটা **index** পজিশন এ **array** এর উপাদান এর মান চেক করতে পারবে

এভাবে:-

```
var friends = ['রহিম', 'কারিম', 'জদু', 'কদু'];  
friends[1] = ('উদু');  
console.log(friends);  
output:-  
['রহিম', 'উদু', 'জদু', 'কদু']
```

১৯. একটা **Array** এর মধ্যে কোন একটা উপাদান এর মান তোমাকে দেয়া আছে এখন সেটার **index** তুমি কিভাবে খুঁজে বের করতে পারো?

আবার আমরা ভ্যালু থেকেও ইন্ডেক্স কে **access** করতে পারি। এর উপায় হলো `arrayName.indexOf(value);`

উদাহরণ:-

```
var friends = ['রহিম', 'কারিম', 'জদু', 'কদু'];  
এই অ্যারে টার "জদু" এর অন্ডেক্স ভ্যালু হবে
```

```
var indexOfJodu = friends.indexOf( "জদু")  
console.log(indexOfJodu);  
output:- ২
```

২০. ধরো, কোন একটা ইন্ডেক্স দিয়ে উপাদান খুঁজতে গেছো। কিন্তু সেটার মান না দিয়ে তোমাকে **undefined** দেখাচ্ছে। সেটা দেখে তুমি কি বুঝবে?

(এটা আমার একান্ত পারসনাল অভিমত। ভুল ও হতে পারে।)

যদি কোন একটা ইন্ডেক্স দিয়ে উপাদান খুঁজতে গিয়ে সেটার মান না পেয়ে **undefined** দেখালে বুঝব ওই উপাদান এর টাইপ **undefined**।

উদাহরণ:

```
var arr = ["hello", undefined, 25];  
console.log(arr[1]);
```

২১. কোন একটা **Array** এর মধ্যে লাস্ট উপাদান হিসেবে কোন উপাদান হিসেবে যোগ করতে চাইলে কিভাবে যোগ করবে। আবার **Array** থেকে শেষের উপাদান টা বের করে দিতে চাইলে কিভাবে বের করে দিবে?

`array.pop()` ইউস করে করা যায়।

এই (`array.pop`) function যেটি শেষ থেকে **delete** হয়েছে সেটি **return** করে।

```
var friends = ['রহিম', 'কারিম', 'জদু', 'কদু'];  
var deleteItem = friends.pop();  
console.log(deleteItem) // 'কদু'
```

২২. কোন একটা **Array** এর মধ্যে প্রথম উপাদান হিসেবে কোন উপাদান হিসেবে যোগ করতে চাইলে কিভাবে যোগ করবে। আবার **Array** থেকে প্রথম উপাদান টা বের করে দিতে চাইলে কিভাবে বের করে দিবে?

`array.shift()` ইউস করে করা যায়।

এই (`array.shift`) function যেটি শুরু থেকে delete হয়েছে সেটি return করে।

```
var friends = ['রহিম', 'কারিম', 'জদু', 'কদু'];
```

```
var deleteltem = friends.shift();
```

```
console.log(deleteltem)// 'রহিম'
```

২৩. তুলনা কিভাবে করতে হয়। এইগুলার মানে কি: `>`, `<`, `==`, `!=`, `<=`, `>=`, `===`, `!==`, `&&`, `||`

১- (`>`) (বড় > ছোট) greater than দুই বা ততোধিক সংখ্যা, পরিমাণ বা মান তুলনা করতে ব্যবহৃত হয়।

2- (`<`) (ছোট < বড়) less than দুই বা ততোধিক সংখ্যা, পরিমাণ বা মান তুলনা করতে ব্যবহৃত হয়।

3- (`==`) double equal হল 'ইকুয়াল টু' চিহ্ন। একে 'তুলনা অপারেটর' বলা হয়। যেমন `5 + 10 == 15`।

4- (`===`) triple equal ও 'তুলনা অপারেটর' কিন্তু এর মানে, ধরন এবং মান উভয়ই সমান হতে হবে।

যেমন: `console.log("1"===1)// true, console.log("1"=== 1)// false`

5- (`!=`) not equal পরীক্ষা করে যে এর দুটি অপারেন্ড সমান নয়, একটি বুলিয়ান ফলাফল প্রদান করে

6- (`<=`) less than equal বাম অপারেন্ড যদি ডান অপারেন্ডের চেয়ে কম বা সমান হয় তবে তার চেয়ে কম বা সমান অপারেটর (`<=`) সত্য প্রদান করে এবং অন্যথায় মিথ্যা

7- (`>=`) grater that equal বাম অপারেন্ড ডান অপারেন্ডের চেয়ে বড় বা সমান হলে বৃহত্তর বা সমান অপারেটর (`>=`) সত্য দেখায় এবং অন্যথায় মিথ্যা।

8- (`!==`) not equal with type ৫ নং টাইপ এর মত কিন্তু এর ধরন এবং মান উভয়ই ভিন্ন হতে হবে।

9- (`&&`) double and দুটি condition এর দুটিই সত্য হতে হবে।

10- (`||`) double or দুটি condition এর যেকোনো একটি সত্য হতে হবে।

২৪. তোমার কাছে: ৮০০০০ টাকার বেশি হলে তুমি **mac** কিনবে, ৬০ টাকার বেশি হলে **gaming** ল্যাপটপ কিনবে, ৪০ হাজার টাকার বেশি হলে **lenovo yoga** কম্পিউটার কিনবে, ২০ হাজার টাকার বেশি হলে পুরান ল্যাপটপ কিনবে। না হয় তুমি মোবাইল দিয়ে কাজ চালাবে।



```
1 var gadgetMoney = 45000;
2 if (gadgetMoney > 80000) {
3   console.log("i would love to buy a mackbook");
4 } else if (gadgetMoney > 60000 && gadgetMoney <= 80000) {
5   console.log("i would love to buy a gaming Pc");
6 } else if (gadgetMoney > 40000 && gadgetMoney <= 60000) {
7   console.log("i would love to buy a lenovo Yoga Pc");
8 } else if (gadgetMoney > 20000 && gadgetMoney <= 40000) {
9   console.log("i would buy a used lapi");
10 } else {
11   console.log("use my phone for work");
12 }
13
```

২৫. আসকে আমার মন ভালো নেই -- এই কথা ৩৯ বার আউটপুট হিসেবে দেখাও।



```
1 for (var i = 1; i <= 39; i++) {
2   console.log("আসকে আমার মন ভালো নেই");
3 }
4
```


২৬. while লুপ কিভাবে কাজ করে ?

একটি while লুপ হল একটি লুপ যেটি চলতে থাকে এবং একটি while স্টেটমেন্ট কার্যকর করে যতক্ষণ পর্যন্ত একটি পূর্বনির্ধারিত শর্ত সত্য থাকে। প্রতিটি পুনরাবৃত্তির পরে, লুপ পরীক্ষা করে যে শর্তটি সত্য রয়েছে। যদি শর্তটি এখন মিথ্যা হয়, লুপটি বন্ধ হয়ে যায়।

২৭. for লুপ কিভাবে কাজ করে ?

১- init ধাপটি প্রথমে এবং শুধুমাত্র একবার সম্পাদিত হয়। এই ধাপটি আপনাকে যেকোনো লুপ কন্ট্রোল ভেরিয়েবল ঘোষণা এবং আরম্ভ করতে দেয়।

২- এর পরে, অবস্থা মূল্যায়ন(condition check) করা হয়। 'ফর' লুপের বডি কার্যকর হওয়ার পরে, নিয়ন্ত্রণের প্রবাহ বৃদ্ধির(incriment) বিবৃতিতে ফিরে যায়। অবস্থা এখন আবার মূল্যায়ন করা হয়।

৩- অবস্থা মূল্যায়ন false ভ্যালু দিলে লুপটি বন্ধ হয়ে যায়।

২৮. while লুপ এর মধ্যে লুপ ভেরিয়েবল চেইঞ্জ না করলে কি সমস্যা হয়।

Infinity times পর্যন্ত লুপ চলতে থাকবে।

২৯. একটা কোড লিখে ৫৮ থেকে ৯৮ পর্যন্ত যত সংখ্যা আছে সেগুলোকে দেখাও



```
1 for (var i = 58; i <= 98; i++) {  
2   console.log(i);  
3 }  
4
```

৩০.একটা কোড লিখে ৪১২ থেকে ৪৫৬ পর্যন্ত যত জোর সংখ্যা আছে সেগুলোকে দেখাও



```
1 for (var i = 412; i <= 456; i += 2) {  
2   console.log(i);  
3 }
```

৩১. একটা কোড লিখে ৫৮১ থেকে ৬২৩ পর্যন্ত যত বিজোড় সংখ্যা আছে সেগুলোকে দেখাও



```
1 for (var i = 581; i <= 623; i += 2) {  
2   console.log(i);  
3 }  
4
```

৩২. while আর for loop এর মধ্যে পার্থক্য কি ?

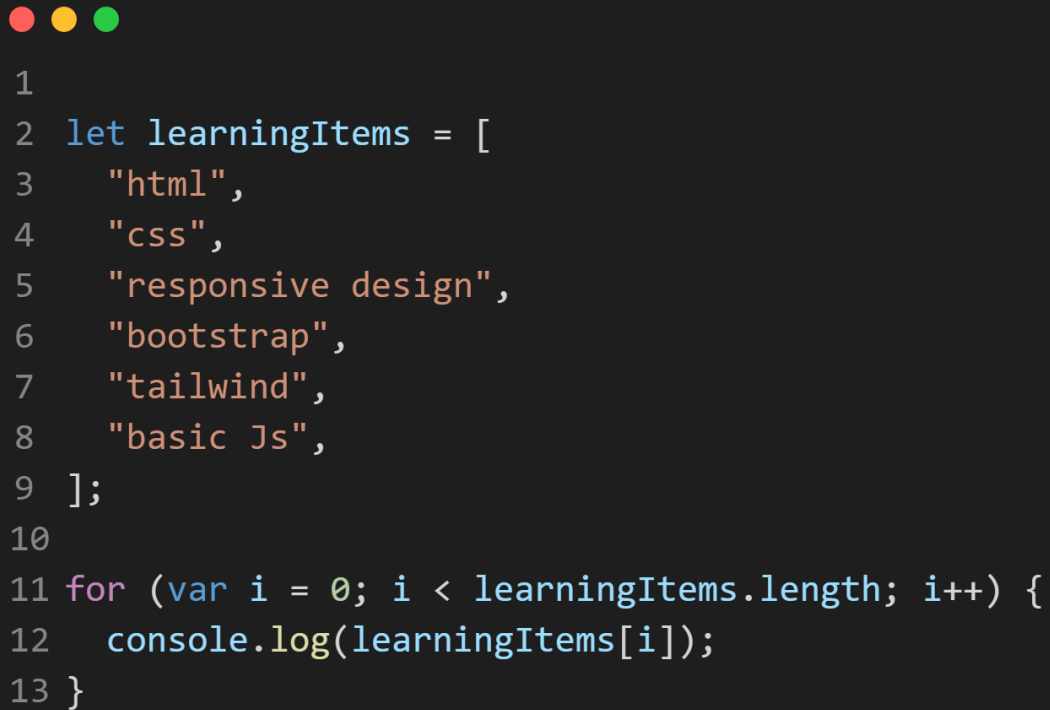
for loop :

- ইনিশিয়ালাইজেশন, কন্ডিশন চেকিং, এবং রিটেশন স্টেটমেন্ট লুপের শুরুতে লেখা হয়।
- এটি শুধুমাত্র তখনই ব্যবহৃত হয় যখন পুনরাবৃত্তির সংখ্যা আগে থেকে জানা যায়।
- শর্তটি 'ফর' লুপে উল্লেখ না থাকলে, লুপটি অসীম সংখ্যক বার পুনরাবৃত্তি করে।
- ইনিশিয়ালাইজেশন শুধুমাত্র একবার করা হয়, এবং এটি পুনরাবৃত্তি হয় নাই।

while loop:

- লুপের শুরুতে ইনিশিয়ালাইজেশন এবং কন্ডিশন চেকিং করা হয়।
- এটি শুধুমাত্র তখনই ব্যবহৃত হয় যখন পুনরাবৃত্তির সংখ্যা জানা যায় না।
- যদি শর্তটি 'while' লুপে উল্লেখ করা না থাকে, তাহলে এটি একটি compilation error এর কারণ হয়।
- কন্ডিশন চেক করার সময় যদি ইনিশিয়ালাইজেশন করা হয়, তাহলে প্রতিবার লুপটি পুনরাবৃত্তি করার সময় ইনিশিয়ালাইজেশন ঘটে।
- iteration statement লুপের ভিতরের যেকোনো বিন্দুতে লেখা যেতে পারে।

৩৩- তুমি এতদিন যা যা জিনিস শিখছো সেগুলার নাম দিয়ে একটা **array** বানাও। তারপর একটা **for** লুপ দিয়ে সেই **array** এর সব উপাদান কে আউটপুট হিসেবে দেখাও।



```
1
2 let learningItems = [
3   "html",
4   "css",
5   "responsive design",
6   "bootstrap",
7   "tailwind",
8   "basic Js",
9 ];
10
11 for (var i = 0; i < learningItems.length; i++) {
12   console.log(learningItems[i]);
13 }
```

৩৪. তুমি এতদিন পর্যন্ত যে যে মডেলের মোবাইল ফোন ইউজ করেছো সেগুলার নাম দিয়ে একটা **array** বানাও। তারপর একটা **while** লুপ দিয়ে সেই **array** এর উপাদান গুলো একটা একটা করে আউটপুট হিসেবে দেখাও



```
1 let mobilesIUsed = [  
2   "Sonny Erricson",  
3   "Nokia 1100",  
4   "Nokia 2600",  
5   "Nokia 5233",  
6   "Walton primo",  
7   "Redmi Note 8",  
8 ];  
9  
10 var count = 0;  
11 while (count < mobilesIUsed.length) {  
12   console.log(mobilesIUsed[count]);  
13   count++;  
14 }  
15
```

৩৫. একটা ফর লুপ চালাও। ৩০ থেকে ৮৬ পর্যন্ত। আর এই লুপ ৪৪ এ গেলে ব্রেক করবে। সেই জিনিস কোড করে দেখাও



```
1 for (var i = 30; i <= 86; i++) {  
2   if (i == 44) {  
3     break;  
4   }  
5   console.log(i);  
6 }  
7
```

৩৬. তোমার যত বই আছে সেগুলার দাম নিয়ে একটা **array** লিখে ফেলো। যে বই গুলোর দাম ২০০ টাকার উপরে সেগুলোকে স্কিপ করবে। অর্থাৎ সেগুলোকে আউটপুট হিসেবে দেখাবে না। বাকিদের কে আউটপুট হিসেবে দেখাবে। দেখা করতে পারো কিনা।



```
1
2 var myBooksPrice = [100, 240, 123, 354, 50, 20, 1200, 150, 160];
3
4 for (var i = 0; i < myBooksPrice.length; i++) {
5     if (myBooksPrice[i] > 200) {
6         continue;
7     }
8     console.log(myBooksPrice[i]);
9 }
10
```