====================

Ajax Intro

====================

118.

# পার্ট 52 Ajax Get Request

<html>

<head>

    <title>DOM Practice</title>

    <!-- Bootstrap CSS -->

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"

  integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

  </head>

<body>

      <button onclick="ExcGetRequest()">ExcGetRequest</button>

    <script>

      function ExcGetRequest(){

        var URL="https://crud.teamrabbil.com/api/v1/ReadProduct";

        var Configuration={method:"GET"};

        fetch(URL,Configuration)

          .then(response=>response.json())

          .then(result=>console.log(result))

          .catch(err=>console.log(err));

      }

    </script>

</body>

</html>

Ans: যখন button এ ক্লিক করব। consol এ অথবা Network এর Response এর ভিতর উত্তর পাব।

119.

**Part 53 Ajax Post Request**

**পার্ট 54 Ajax Post Request**

<html>

<head>

    <title>DOM Practice</title>

    <!-- Bootstrap CSS -->

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"

  integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

  </head>

<body>

      <button onclick="ExecPostRequest()">Exec Post Request</button>

    <script>

      function ExecPostRequest(){

        var URL="https://crud.teamrabbil.com/api/v1/CreateProduct"

        var BodyData={Img:"123",ProductCode:"123",ProductName:"123",Qty:"123",TotalPrice:"123",UnitPrice:"123"};

        var Config={

          method:"POST",

          headers:{'Accept':'application/json','Content-Type':'application/json'},

          data:JSON.stringify(BodyData)

        };

        fetch(URL,Config)

          .then(response=>response.json())

          .then(result=>console.log(result))

          .catch(err=>console.log(err));

      }

    </script>

</body>

</html>

Ans: যখন button এ ক্লিক করব। consol এ অথবা Network এর Response এর ভিতর উত্তর পাব।

====================================

**JavaScript Tutorial Bangla ES6**

====================================

\*\* node.js Install করব এর মাধ্যমে result বের করব। vs code এ যাব, terminal এ result দেখতে পারব। তার জন্য node এর পর file এর নাম লিখব। যেমনঃ node main.js এখানে main.js হল file এর নাম

1.

**hello world Part 02**

console.log('Hello Word');

Ans: Hello Word

2.

**Strict Mode part 04**

"use strict";

Myname();

function Myname(){

name = "rabbil hasan";

}

Ans: error দেখাবে।

3.

**Spread Operator Part 05**

let poorCountry=['bangladesh','srilanka'];

let richCountry=[...poorCountry,'America','Canada','Italy'];

console.log(richCountry);

Ans: [ 'bangladesh', 'srilanka', 'America', 'Canada', 'Italy' ]

4.

**Without Spread Operator Part 06**

let poorCountry=['bangladesh','srilanka'];

let richCountry=['America','Canada','Italy'];

richCountry.push(poorCountry);

console.log(richCountry);

Ans: [ 'America', 'Canada', 'Italy', [ 'bangladesh', 'srilanka' ] ]

5.

**Rest parameter Part 07**

function Calculation(...numbers){

    let sum=0;

    for(let i of numbers){

        sum=sum+i;

    }

    console.log(sum);

}

Calculation(1,2,3,4,5,8,9)

Ans: 32

6.

function Calculation(a,b,...numbers){

    let sum=0;

    for(let i of numbers){

        sum=sum+i;

    }

    console.log(sum);

}

Calculation(10,20,1,2,3,4)

Ans: 10.

এখানে ১০,২০ এই দুটি সংখ্যা যোগ করা হয় নাই। তাই উত্তর আসবে ১০

7.

**Dynamic Function Part 08**

var name = function(nameValue){

    return nameValue;

}

console.log(name("Rabbil Hasan"))

Ans: Rabbil Hasan

8.

**ES6 Variable Part 09**

var name = "Rabbil";

name="Rain"; // Reasign

console.log(name);

Ans: Rain

\*\*একই variable এর নাম এর মধ্যে নতুন value যোগ করা। এটাই হচ্ছে Reasign.

\*\*Var ও let ব্যাবহার করে Reasign করা যায়। কিন্তু const ব্যাবহার করে Reasign করা যায় না।

9.

let name = "Rabbil";

name="Rain";   // Reasign

console.log(name);

Ans: Rain

10.

const name = "Rabbil";

name="Rain";   // Reasign

console.log(name);

Ans: error আসবে। কারন const দিয়ে Reasign করা যায় না।

11.

var name = "Rabbil";

var name="Rain";   // Redeclear

console.log(name);

Ans: Rain

\*\*Redeclear এর মানে একই variable, var দিয়ে একবার declear করার পর।

আবার var দিয়ে declear করা।

\*\*Var দিয়ে Redeclear করা যায়। কিন্তু let, const দিয়ে Redeclear করা যায় না।

12.

let name = "Rabbil";

let name="Rain";   // Redeclear

console.log(name);

Ans: : error আসবে। কারন let দিয়ে Redeclear করা যায় না।

13.

const name = "Rabbil";

const name="Rain";   // Redeclear

console.log(name);

Ans: : error আসবে। কারন const দিয়ে Redeclear করা যায় না।

14.

**Variable Scope Part 10**

**Global Scope:** এর মানে হচ্ছে এমন জায়গায় এমন ভাবে variable declare করব। ঐ variable টা যেখানে থেকে খুশি ব্যাবহার করতে পারব।এমন হলে সেই variable কে Global Scope বলি।

**Local Scope:** এর মানে হচ্ছে variable টা এমন ভাবে declare করব, সেটা যেখানে declare করেছি সেখানে ব্যাবহার করা যাচ্ছে অন্য কোথায়ও ব্যাবহার করা যাচ্ছে না তখন ঐ ধরনের Scope কে Local Scope বলে।

function myFun(){

    var name="ECMAScript6";

    console.log(name);

}

myFun();

Ans: ECMAScript6

\*\*এখানে name variable টা হল local scope. কারন function এর মধ্যে অউটপুট দেখানো হইছে। function এর বাইরে অউটপুট দেখানো হলে error আসত।

15.

var name="ECMAScript6";

function myFun(){

    console.log(name);

}

myFun();

Ans: ECMAScript6

\*\*এখানে name variable টা হল global scope. কারন function এর মধ্যে অউটপুট দেখানো হইছে। function এর বাইরে অউটপুট দেখানো হলেও একই উত্তর আসবে কোন error আসবে না। যেখানে ই ব্যাবহার করি না কেন কোন অউটপুট error আসবে না। তাই global scope.

16.

var name="ECMAScript6";

function myFun(){

}

console.log(name);

Ans: ECMAScript6

\*\*যখন variable সব জায়গাতে কাজ করে, function এর বাইরে এবং ভিতরে সব জায়গাতে কাজ ক্রে,এই ধরনের function কে global scope বলে।

17.

**Variable Hoisting Part 11**

**Hoisting=** মানে উত্তোলন

\*\*Value assign হবে আগে, অর্থাৎ value টাকে আমরা উত্তোলন করব আগে Declare করব পরে।

\*\*যদি value টাকে আমরা assign করেছি আগে, variable কে Declare করেছি পরে। এই বিষয় টাকে বলা হয় hoisting

name="Rabbil Hasan";

console.log(name);

var name;

Ans: Rabbil Hasan

18.

**Simple For Loop Part 12**

var i;

for(i=0;i<=10;i++){

    console.log("Allah");

}

Ans:

Allah

Allah

Allah

Allah

Allah

Allah

Allah

Allah

Allah

Allah

Allah

19.

var i;

for(i=0;i<=100;i=i+30){

    console.log("Allah");

}

Ans:

Allah

Allah

Allah

Allah

20.

**Simple For of loop Part 13**

এই **For of loop** টা কিছুটা foreach loop এর মত কাজ করে।

var MyCountry=['Dhaka','Rangpur','Khulna','Barisal'];

for(let MyCity of MyCountry){

    console.log(MyCity);

}

Ans:

Dhaka

Rangpur

Khulna

Barisal

21.

**JavaScript ES6 Object Part 14**

\*\*যা কিছু আছে সেগুলো object যেমনঃ Bill gates হল object.

\*\*Bill gates এর বৈশিষ্ট গুলো হচ্ছে এই object এর property.

\*\*একটা বিষয় এর ভিতরে, জিনিসের ভিতরে অথবা একটা বস্তুর ভিতরে, আমরা যখন বৈশিষ্ট গুলো খুজতে যাই, তখন বৈশিষ্ট গুলো বলা হয় ঐ object এর property। আর পুরা বিষয়কে বলা হয় object ঐ বৈশিষ্ট এর।

var billGates={

    shirt:true,

    shirtColor:"offWhite",

    SwaterColor:"green"

}

\*\*এখানে billGates হল object. billGates এর মধ্যে যা আছে সেগুলো হল property.

\*\* object এর মধ্যে object থাকলে সেটা হল **Nested** object

const person = {

name: "Alice",

age: 30,

address: {

street: "123 Main St",

city: "Anytown",

zipCode: "12345"

},

};

\*\*এখানে address হল Nestedobject

22.

**JavaScript ES6 Object Part 15**

var billGates={

    shirt:true,

    shirtColor:"offWhite",

    smile:true,

    SwaterColor:"green"

}

console.log(billGates['shirtColor']);

Ans: offWhite

23.

var billGatesPro={

    shirt:{

    color:"offWhite",

    Quality:"Good",

    price:"200usd",

},

swatter:{

    color:"Gary",

    Quality:"Good",

    price:"230USD",

    warm:"best"

},

face:{

    similing:"Yes",

    chosma:"Yes",

    teeth:"white,Big"

    }

}

console.log(billGatesPro['swatter']['price']);

Ans: 230USD

24.

**ES6 For in loop 16**

Property কে props বলা হয়

var billGates={

    shirt:true,

    shirtColor:"offWhite",

    smile:true,

    SwaterColor:"gray",

    chosma:true

}

for(let props in billGates){

    console.log(props);

}

Ans:

shirt

shirtColor

smile

SwaterColor

chosma

25.

var billGates={

    shirt:true,

    shirtColor:"offWhite",

    smile:true,

    SwaterColor:"gray",

    chosma:true

}

for(let props in billGates){

    console.log(billGates[props]);

}

Ans:

true

offWhite

true

gray

true

26.

var billGates={

    shirt:true,

    shirtColor:"offWhite",

    smile:true,

    SwaterColor:"gray",

    chosma:true

}

for(let props in billGates){

    console.log(props+"= "+billGates[props]);

}

Ans:

shirt= true

shirtColor= offWhite

smile= true

SwaterColor= gray

chosma= true

27.

**Decision Making Part 17**

var billGates={

    shirt:true,

    shirtColor:"offWhite",

    smile:true,

    SwaterColor:"gray",

    chosma:true

}

if(billGates['shirtColor']=="red"){

    console.log("Shirt Color is Red");

}

else if(billGates['shirtColor']=="blue"){

     console.log("Shirt Color is blue");

}

else{

    console.log("Shirt Color Not Found");

}

Ans: Shirt Color Not Found

28.

**Simple Function And Parameterized functions part 18**

**Simple Function**

function myName(){

    var x=10;

    var y=10;

    var z=x+y;

    console.log(z);

}

myName();

Ans: 20

29.

**\*\*Parameterized functions**

function myName(x,y){

    var z = x+y;

    console.log(z);

}

myName(10,20);

Ans: 30

30.

**Rest Parameters Functions Part 19**

\*\*Rest parameter টা array আকারে value receive করে। তিনটি ডট(…) থাকে কোন parameter এর মধ্যে তাহালে সেটাই Rest parameter.

\*\*Rest parameter এক সাথে অনেক গুলো value receive করে। অর্থাৎ Rest parameter এর কাজ হল একটা parameter দিয়ে array আকারে অনেক গুলো value কে এক সাথে receive করা।

function something(...x){

    console.log(x);

}

something(1,2,3,7,8,9,6,"A","B","C")

Ans: [1, 2, 3, 7, 8, 9, 6, 'A', 'B', 'C']

31.

function something(...x){

    console.log(x[6]);

}

something(1,2,3,7,8,9,5,6,"A","B","C")

Ans: 5

32.

**Function Returns Part 20**

function MyFunction(){

    return "Bangladesh";

}

console.log(MyFunction())

Ans: Bangladesh

33.

function FunOne(){

    return 20;

}

function FunTwo(){

    return 30;

}

console.log(FunOne()+FunTwo());

Ans: 50

34.

function FunOne(){

    return 20;

}

function FunTwo(){

    return FunOne();

}

console.log(FunTwo());

Ans: 20

35.

**Anonymous functions Part 21**

var myFun=function(){

    return "Hello";

}

console.log(myFun());

Ans: Hello

36.

var myFun=function(x){

    return x;

}

console.log(myFun("Hello"));

Ans: Hello

37.

var myFun=function(...x){

    return x;

}

console.log(myFun(1,2,3,4,5,6,7));

Ans: [1, 2, 3, 4, 5, 6, 7]

38.

\*\*Anonymous functions টা reassign করা যায়। Anonymous functions হল নাম ছাড়া function,যার কোন নাম নাই।

\*\*একই function বার বার লিখা যায়। শেষে যে function টা লিখব তার উত্তর পাব, আগের function এর কোন উত্তর পাব না।

var myFun=function(){

    return "Hello";

}

var myFun=function(){

    return "Hello Again";

}

console.log(myFun());

Ans: Hello Again

39.

**Arrow Function Part 22**

\*\*Arrow Function ছোট syntax এর মাধ্যমে লিখা যায়। Function এর নাম ও Function keyword এর প্রয়োজন হয় না।

\*\* Arrow Function এক ধরনের Anonymous functions. তাই Anonymous functions যে যে বৈশিষ্ট্য গুলো ধারণ করে এই Arrow Function ও সেই সেই বৈশিষ্ট্য ধারণ করে।

\*\* Arrow Function এ Constructor function ব্যাবহার করা যাবে না।

()=>{} এটা Arrow Function এর symbol, এভাবে Arrow Function ব্যাবহার করা হয়।

var myArrow = ()=>{

    console.log("My first Arrow function");

}

myArrow();

Ans: My first Arrow function

40.

var myArrow = (x)=>{

    console.log(x);

}

myArrow("My first Arrow function");

Ans: My first Arrow function

41.

var myArrow = (...x)=>{

    console.log(x);

}

myArrow(1,2,3,4,5,6);

Ans: [ 1, 2, 3, 4, 5, 6 ]

42.

var myArrow = ()=>{

    return "Hello World";

}

console.log(myArrow())

Ans: Hello World

43.

**ES6 Arrays Part 23**

শুধু আলোচনা

44.

**Simple Arrays Part 24**

var myArray=["A","B","C","D","E"];

var myArrayPro= new Array("A","B","C","D","E");

console.log(myArray[1]);

Ans: B

45.

var myArray=["A","B","C","D","E"];

var myArrayPro= new Array("A","B","C","D","E");

for(let item of myArray){

    console.log(item);

}

Ans:

A

B

C

D

E

46.

var myArray=["A","B","C","D","E"];

var myArrayPro= new Array("A","B","C","D","E");

for(let item of myArrayPro){

    console.log(item);

}

Ans:

A

B

C

D

E

47.

**Multidimensional Arrays Part 25**

var bangladesh=["Dhaka","Rangpur","Ctg"];

var india=["Mumbai","Chennai","kolkata"];

var china=["Uhan","Shanghai","Hongkong"];

var asia= [bangladesh,india,china]

    console.log(asia[1][2]);

Ans: kolkata

48.

**Array de structuring Part 26**

\*\*de structuring এর মানেStructure কে ভেংগে ভেংগে সেটাকে টুকরা টুকরা part করে ফেলা।

\*\* যখন আমাদের array অনেক বেশি complex আকার ধারণ করে, বা array এর বিষয় গুলো সেই বিষয় গুলো অনেক বেশি complex হয়ে যায়, তখন আমরা array কে de structuring করে সেই array কে ভেংগে ভেংগে আমরা part by part আলাদা করে ফেলি। মুলত complex array কে ভেংগে আলাদা করে ফেলার যে বিষয়টা এই বিষয়টাই de structuring.

var countries=["BAN","IND","CANA","PAK"];

var[a,,,b]=countries

console.log(b);

Ans: PAK

49.

var countries=["BAN","IND","CANA","PAK"];

var[a,c,d,b]=countries

console.log(c);

Ans: IND

50.

var countries=["BAN","IND","CANA","PAK"];

var[ , ,a]=countries

console.log(a);

Ans: CANA

51.

**ES6 Map Part 27**

Map হল group of data অথবা collection of data, যে collection এর ভিতরে বা গ্রুপ এর ভিতরে, প্রত্যেকটা data এর জন্য আলাদা আলাদা key থাকে। key এবং data এই দুটোকে বিষয়কে এক সাথে বলা হয় key ও value pairing. Pairing এর মানে যুগল। তার মানে Map এর মধ্যে যখন আমরা data রাখি, data গুলো যুগল অবস্থায় থাকে। সেই যুগল অথবা জোড়া এর মধ্যে একটা থাকে key আর একটা থাকে value.

\*\* new Map() এটা map এর object.

var myMap= new Map();

myMap.set("key1","Bangladesh")

myMap.set("key2","India")

myMap.set("key3","Sirilanka")

myMap.set("key4","Maldiv")

myMap.set("key5","Bhutan")

myMap.set("key6","Nepal")

myMap.set("key7","Thailand")

console.log(myMap.values())

Ans:

{

'Bangladesh',

'India',

'Sirilanka',

'Maldiv',

'Bhutan',

'Nepal',

'Thailand'

}

এখানে object আকারে উত্তর আসবে । অথবা array আকারে return করবে।

52.

var myMap= new Map();

myMap.set("key1","Bangladesh")

myMap.set("key2","India")

myMap.set("key3","Sirilanka")

myMap.set("key4","Maldiv")

myMap.set("key5","Bhutan")

myMap.set("key6","Nepal")

myMap.set("key7","Thailand")

console.log(myMap.keys())

Ans:

{

'key1',

'key2',

'key3',

'key4',

'key5',

'key6',

'key7'

}

এখানে object আকারে উত্তর আসবে । অথবা array আকারে return করবে।

53.

**ES6 Map Part 28**

var myMap= new Map();

myMap.set("key1","Bangladesh")

myMap.set("key2","India")

myMap.set("key3","Sirilanka")

myMap.set("key4","Maldiv")

myMap.set("key5","Bhutan")

myMap.set("key6","Nepal")

myMap.set("key7","Thailand")

for(let myValue of myMap.values()){

    console.log(myValue);

}

Ans:

Bangladesh

India

Sirilanka

Maldiv

Bhutan

Nepal

Thailand

54.

var myMap= new Map();

myMap.set("key1","Bangladesh")

myMap.set("key2","India")

myMap.set("key3","Sirilanka")

myMap.set("key4","Maldiv")

myMap.set("key5","Bhutan")

myMap.set("key6","Nepal")

myMap.set("key7","Thailand")

for(let myKey of myMap.keys()){

    console.log(myKey);

}

Ans:

key1

key2

key3

key4

key5

key6

key7

55.

**ES6 Map Part 29**

\*\*Delete(key) এর কাজ key ধরে map এর কোন একটা জোড়াকে delete করে দিতে পারে।

\*\*get() এর কাজ কোন specific key ধরে সেই key এর value কে বের করে নিয়ে আসতে পারে।

\*\*clear|() এর কাজ map এর ভিতর যতগুলো জোড়া আছে বা যতগুলো pair আছে সব গুলকে এক বারে clear করে নেয়। মানে delete করে দিবে সব।

\*\* has(key) এর কাজ map এর ভিতরে অথবা map এর collection এর ভিতরে, specifc কোন value আছে কিনা সেই বিষয়টাকে check করে।

var myMap= new Map();

myMap.set("key1","Bangladesh")

myMap.set("key2","India")

myMap.set("key3","Sirilanka")

myMap.set("key4","Maldiv")

myMap.set("key5","Bhutan")

myMap.set("key6","Nepal")

myMap.set("key7","Thailand")

myMap.delete("key1")

myMap.delete("key2")

for(let myValue of myMap.values()){

    console.log(myValue);

}

Ans:  
  
Sirilanka

Maldiv

Bhutan

Nepal

Thailand

56 .  
var myMap= new Map();

myMap.set("key1","Bangladesh")

myMap.set("key2","India")

myMap.set("key3","Sirilanka")

myMap.set("key4","Maldiv")

myMap.set("key5","Bhutan")

myMap.set("key6","Nepal")

myMap.set("key7","Thailand")

myMap.clear()

for(let myValue of myMap.values()){

    console.log(myValue);

}

Ans: clear() এর কাজ map এর ভিতর যা কিছু আছে delete করে দিবে। তাই কোন উত্তর আসে নাই।

57.

var myMap= new Map();

myMap.set("key1","Bangladesh")

myMap.set("key2","India")

myMap.set("key3","Sirilanka")

myMap.set("key4","Maldiv")

myMap.set("key5","Bhutan")

myMap.set("key6","Nepal")

myMap.set("key7","Thailand")

console.log(myMap.get("key6"));

Ans: Nepal

58.   
  
var myMap= new Map();

myMap.set("key1","Bangladesh")

myMap.set("key2","India")

myMap.set("key3","Sirilanka")

myMap.set("key4","Maldiv")

myMap.set("key5","Bhutan")

myMap.set("key6","Nepal")

myMap.set("key7","Thailand")

if(myMap.has("key1")){

    console.log("Yes");

}else{

    console.log("No");

}

Ans: Yes  
  
  
59.

var myMap= new Map();

myMap.set("key1","Bangladesh")

myMap.set("key2","India")

myMap.set("key3","Sirilanka")

myMap.set("key4","Maldiv")

myMap.set("key5","Bhutan")

myMap.set("key6","Nepal")

myMap.set("key7","Thailand")

if(myMap.has("key10")){

    console.log("Yes");

}else{

    console.log("No");

}

Ans: No

60.

**ES6 Set Part 30**

\*\* set ও map এর মত data Collection করে রাখে।

\*\* map এর যেমন কাজ set এর কাজও সে রকম। তবে set এ map এর থেকে একটু ভিন্নতা আছে, সেই ভিন্নতা হচ্ছে, map এর মধ্যে চাইলে duplicate value রাখতে পারব, মানে একই value বার বার map এর মধ্যে রাখতে পারব কোন সমস্যা হবে না। কিন্তু set এর মধ্যে duplicate value রাখতে পারব না। যদি duplicate value রাখিও set সেটা count করবে না।

\*\* set এর মধ্যে duplicate value রাখতে পারব না।  
  
var mySet= new Set();

mySet.add('Bangladesh');

mySet.add('India');

mySet.add('Nepal');

mySet.add('Bhutan');

mySet.add('Maldiv');

mySet.add('Chaina');

console.log(mySet);

Ans: { 'Bangladesh', 'India', 'Nepal', 'Bhutan', 'Maldiv', 'Chaina' }

61.

var mySet= new Set();

mySet.add('Bangladesh');

mySet.add('India');

mySet.add('Nepal');

mySet.add('India');

mySet.add('Bhutan');

mySet.add('Maldiv');

mySet.add('Chaina');

mySet.add('India');

console.log(mySet);

Ans: { 'Bangladesh', 'India', 'Nepal', 'Bhutan', 'Maldiv', 'Chaina' }

এখানে 'India' value টা কয়েকবার দেয়া আছে, মানে duplicate value দেয়া আছে। set এর নিয়ম অনযায়ী যতবার ই duplicate value দেই না কেন একবার ই value পাব।

62.

var mySet= new Set(['A','B','C','D']);

console.log(mySet);

Ans: { 'A', 'B', 'C', 'D' }

63.

var mySet= new Set(['A','A','B','C','D']);

console.log(mySet);

Ans: { 'A', 'B', 'C', 'D' }

64.

**ES6 Set Part 31**

.clear() = set এর মধ্যে যতগুলো value থাকবে সব গুলো value clear হয়ে যাবে।

.delete(value) = specific ভাবে কোন value delete করার জন্য .delete(value) এটা ব্যাবহার করব।

.has() = কোন value আছে কিনা তা check করার জন্য .has() এটা ব্যাবহার করব।  
.values() = সবগুলো value একবারে বের করে নিতে আসতে চাইলে values()এটা ব্যাবহার করব।

Set.size = আমরা যদি set এর size নির্ণয় করতে চাই, সে ক্ষেত্রে Set.size লিখলেই ঐ set এর যে size অর্থাৎ কতগুলো element set এর মধ্যে আছে সে সংখ্যাটা চলে আসবে।  
  
  
var mySet= new Set();

mySet.add('Bangladesh');

mySet.add('India');

mySet.add('Nepal');

mySet.add('India');

mySet.add('Bhutan');

mySet.add('Maldiv');

mySet.add('Chaina');

mySet.add('India');

mySet.clear();

console.log(mySet);

Ans: এখানে কোন উত্তর আসবে না কারন clear() এটা ব্যাবহার করেছি।

65 .  
  
var mySet= new Set();

mySet.add('Bangladesh');

mySet.add('India');

mySet.add('Nepal');

mySet.add('India');

mySet.add('Bhutan');

mySet.add('Maldiv');

mySet.add('Chaina');

mySet.add('India');

mySet.delete('Nepal');

console.log(mySet);

Ans: { 'Bangladesh', 'India', 'Bhutan', 'Maldiv', 'Chaina' }

এখানে 'Nepal' delete করা হয়েছে।

66.

var mySet= new Set();

mySet.add('Bangladesh');

mySet.add('India');

mySet.add('Nepal');

mySet.add('India');

mySet.add('Bhutan');

mySet.add('Maldiv');

mySet.add('Chaina');

mySet.add('India');

console.log(mySet.size);

Ans: 6

67.

var mySet= new Set();

mySet.add('Bangladesh');

mySet.add('India');

mySet.add('Nepal');

mySet.add('India');

mySet.add('Bhutan');

mySet.add('Maldiv');

mySet.add('Chaina');

mySet.add('India');

console.log(mySet.values());

Ans:

{

'Bangladesh',

'India',

'Nepal',

'Bhutan',

'Maldiv',

'Chaina'

}  
  
  
68.

var mySet= new Set();

mySet.add('Bangladesh');

mySet.add('India');

mySet.add('Nepal');

mySet.add('India');

mySet.add('Bhutan');

mySet.add('Maldiv');

mySet.add('Chaina');

mySet.add('India');

if(mySet.has('Maldiv')){

    console.log('Yes');

}else{

    console.log('No');

}

Ans: Yes

69.

var mySet= new Set();

mySet.add('Bangladesh');

mySet.add('India');

mySet.add('Nepal');

mySet.add('India');

mySet.add('Bhutan');

mySet.add('Maldiv');

mySet.add('Chaina');

mySet.add('India');

if(mySet.has('Canada')){

    console.log('Yes');

}else{

    console.log('No');

}

Ans: No

70.

**ES6 Class Part 32**

শুধু আলোচনা

71.  
**Create Class & Object Part 33**

class MyClass{

    MyFun(){

        console.log('Hello I am from function inside class')

    }

      MyFun1(){

        console.log('Hello I am from function inside class1')

    }

      MyFun2(){

        console.log('Hello I am from function inside class2')

    }

      MyFun3(){

        console.log('Hello I am from function inside class3')

    }

}

var obj = new MyClass;

obj.MyFun();

Ans: Hello I am from function inside class

72.

class MyClass{

    MyFun(){

        console.log('Hello I am from function inside class')

    }

      MyFun1(){

        console.log('Hello I am from function inside class1')

    }

      MyFun2(){

        console.log('Hello I am from function inside class2')

    }

      MyFun3(){

        console.log('Hello I am from function inside class3')

    }

}

var obj = new MyClass;

obj.MyFun();

obj.MyFun1();

obj.MyFun2();

obj.MyFun3();

Ans:

Hello I am from function inside class

Hello I am from function inside class1

Hello I am from function inside class2

Hello I am from function inside class3  
  
  
  
  
  
73.  
  
class MyClass{

    MyFun(name){

        console.log(name)

    }

      MyFun1(name){

        console.log(name)

    }

      MyFun2(name){

        console.log(name)

    }

      MyFun3(name){

        console.log(name)

    }

}

var obj = new MyClass;

obj.MyFun("Bangladesh");

obj.MyFun1("India");

obj.MyFun2("Maldiv");

obj.MyFun3("Nepal");

Ans:

Bangladesh

India

Maldiv

Nepal

74.

**Part 34 Class Constructor**

\*\*Constructor হল class এর নিজস্ব মেথড অথবা ফাংশন।

\*\* ক্লাস এর অবজেক্ট তৈরি হলেই, constructor অটো ক্ল হয়।

\*\* Constructor function রিটার্ন করতে পারে না।

\*\* একটা class এর মধ্যে একটা Constructor function থাকবে। একটা class এর মধ্যে একাদিক Constructor function তৈরি করা যাবে না।

class MyClass{

    constructor(){

        console.log("Hello I am constructor");

    }

}

new MyClass();

Ans: Hello I am constructor

75.

**Class Constructor 35**

class MyClass{

    constructor(a,b){

        console.log(a+b);

    }

}

new MyClass(10,20);

Ans: 30

76.

class MyClass{

    constructor(a,b){

        this.firstNum=a

        this.secondNum=b

    }

    add(){

        let result = this.firstNum + this.secondNum;

           console.log(result);

    }

}

var obj = new MyClass(10,20);

obj.add();

Ans: 30

এখানে this.firstNum ও this.secondNum এই দুটি হল class variable, আমরা class এর মধ্যে যে কোন জায়গাতে class variable ব্যাবহার করতে পারি।  
  
  
77.

**Static Keyword 36**

class MyClass{

    hello(){

        console.log('Hello World');

    }

}

var obj = new MyClass();

obj.hello();

Ans: Hello World

78.

class MyClass{

    static hello(){

        console.log('Hello World');

    }

}

 MyClass.hello();

Ans: Hello World

এখানে class এর মধ্যে static ব্যাবহার করা হইছে। তাই object তৈরি করে call করা লাগে নাই, সরাসরি class এর মাধ্যমে call করা হইছে। static থাকলে সরাসরি class দিয়ে কল করব।

79.

**Part 37 ES6 Class Inheritance**

**\*\* Inheritance** এর মানে উত্তরাধিকার

class parent{

    hello1(){

        console.log('Hello 1 Function');

    }

    hello2(){

        console.log('Hello 2 Function');

    }

}

class child extends parent{

}

var obj=new child();

obj.hello1();

obj.hello2();

Ans:

Hello 1 Function

Hello 2 Function

80.

**Part 38 ES6 Class Inheritance**

class parent{

    hello1(){

        console.log('Hello 1 Function');

    }

    hello2(){

        console.log('Hello 2 Function');

    }

}

class child extends parent{

}

var obj=new parent();

obj.hello1();

obj.hello2();

Ans:

Hello 1 Function

Hello 2 Function

81.

বাবার কোন function কে ছেলে যদি পরিবর্তন করে বা সেই function এর কার্যক্রম বা আচরণ পরিবর্তন করে তখন তাকে function overriding বা method overriding বলে। এটা কিন্তু ছেলের একটা অতিরিক্ত ক্ষমতা। বাবার property বাবা ব্যাবহার করতে পারে, চাইলেই বাবা পরিবর্তন করতে পারে না। কিন্তু ছেলে বাবার property ব্যাবহার করতে পারে ও চাইলেই বাবা পরিবর্তন করতে পারে। এই অতিরিক্ত ক্ষমতা ছেলের আছে কিন্তু বাবার নাই।

class parent{

    hello1(){

        console.log('Hello 1 Function');

    }

    hello2(){

        console.log('Hello 2 Function');

    }

}

class child extends parent{

     hello1(){

        console.log('Hello 1 Function Hello 1 Function Hello 1 Function child');

    }

}

var obj=new child();

obj.hello1();

Ans: Hello 1 Function Hello 1 Function Hello 1 Function child

82.

**Part 41 ES6 Modules**

\*\*Named = যে নাম ধরে রপ্তানি করা হচ্ছে সেই নাম ধরে আমদানি করা হচ্ছে সেটা হল Named.

\*\*Default = যে নাম ধরে রপ্তানি করা হচ্ছে সেই নাম ধরে আমদানি করা হচ্ছে না, অন্য নাম ধরে আমদানি করা হচ্ছে সেটা হল Default.

83.

**Part 39 Super Keyword**

Super Keyword = এর কাজ child class এর মধ্যে parent class এর মেথডকে call করতে পারব। এর জন্য Super Keyword ব্যাবহার করা হয়।

class parent{

    hello1(){

        console.log('Hello 1 Function');

    }

    hello2(){

        console.log('Hello 2 Function');

    }

}

class child extends parent{

     demo(){

        super.hello1();

        super.hello2();

    }

}

var obj=new child();

obj.demo();

Ans:

Hello 1 Function

Hello 2 Function

84.  
  
**Part 40 ES6 Modules**

Module = জোট, একত্রিত কিছু

শুধু আলোচনা

85.  
  
**Part 42 ES6 Modules Export Import**

Javascript এর জন্য সবচেয়ে ভাল code editor হচ্ছে webstrom  
  
China.js:

const *laptop* = "Asus Laptop, Hp Laptop";  
export { *laptop* };

Bangladesh.js:

import {*laptop*} from './China.js';  
  
*console*.log(*laptop*);

index.html:

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Title</title>  
 <script type="module" src="./Bangladesh.js"></script>  
</head>  
<body>  
  
</body>  
</html>

Ans: Asus Laptop, Hp Laptop

86.

**Part 43 ES6 Modules Export Import**

China.js:

const *laptop* = "Asus Laptop, Hp Laptop";  
  
function MobilePhn(){  
 *console*.log("Mobile Phn From China");  
}  
export { *laptop*, MobilePhn};

Bangladesh.js:

import {*laptop*,MobilePhn} from './China.js';  
  
MobilePhn();

index.html:

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Title</title>  
 <script type="module" src="./Bangladesh.js"></script>  
</head>  
<body>  
  
</body>  
</html>

Ans: Mobile Phn From China

87.

China.js:

const *laptop* = "Asus Laptop, Hp Laptop";  
  
function MobilePhn(){  
 *console*.log("Mobile Phn From China");  
}  
  
function Cosmetic(){  
 *console*.log("Cosmetic From China");  
}  
  
export { *laptop*, MobilePhn, Cosmetic};

Bangladesh.js:  
  
import {*laptop*,MobilePhn,Cosmetic} from './China.js';  
  
MobilePhn();  
Cosmetic();

index.html:

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Title</title>  
 <script type="module" src="./Bangladesh.js"></script>  
</head>  
<body>  
  
</body>  
</html>

Ans:

Mobile Phn From China

Cosmetic From China

88.

**Part 44 ES6 Modules Export Import**

China.js:

class China {  
  
 Toys(){  
 alert("Toys Imported From China");  
 }  
}  
  
export { China };

Bangladesh.js:  
  
import {China} from './China.js';  
  
let obj = new China();  
  
obj.Toys();

index.html:

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Title</title>  
 <script type="module" src="./Bangladesh.js"></script>  
</head>  
<body>  
  
</body>  
</html>

Ans: Toys Imported From China

Alert আসবে Toys Imported From China দিয়ে।

89.

**Part 45 ES6 Modules Export Import**

China.js:

class Cow {  
  
 Toys(){  
 alert("Toys Imported From China");  
 }  
}  
  
export default Cow ;

Bangladesh.js:

import Goro from './China.js';  
  
let obj = new Goro();  
  
obj.Toys();

এখানে Cow নামে export করেছি, আর Goro নামে import করেছি।

index.html:

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Title</title>  
 <script type="module" src="./Bangladesh.js"></script>  
</head>  
<body>  
  
</body>  
</html>

Ans: Toys Imported From China

Alert আসবে Toys Imported From China দিয়ে।