

Java script

اعداد//أسامة زهير النداوي
للتواصل//osamah.zuhair@yahoo.com

أهم المحطات في تاريخ JavaScript:

1. 1995 - إنشاء JavaScript

تم تطويرها تحت اسم Mocha، ثم تغيير الاسم إلى LiveScript، وأخيرًا إلى JavaScript بعد شراكة بين Netscape و Sun Microsystems، لجذب انتباه المبرمجين المهتمين بلغة Java.

2. 1996 - اعتماد JavaScript كلغة قياسية

قامت Microsoft بإنشاء نسخة خاصة بها تسمى JScript لتعمل على متصفح Internet Explorer.

دخلت JavaScript في عملية التقييس من خلال ECMA International، وهي منظمة تحدد المعايير التكنولوجية.

3. 1997 - إطلاق المواصفة الرسمية (ES1) ECMAScript

تمت الموافقة على (ES1) ECMAScript 1 كأول معيار رسمي للغة.

4. 2009 - الثورة مع ECMAScript 5 (ES5)

أضافت ES5 ميزات جديدة، مثل JSON و strict mode، مما حسن الأداء والاستقرار.

5. 2015 - القفزة الكبرى مع ECMAScript 6 (ES6)

قدمت ES6 (ECMAScript 2015) ميزات حديثة، مثل let/const، ال arrow functions، ال classes، ال modules، مما جعل JavaScript أكثر قوة وشبيهة باللغات الكائنية مثل Java و #C.

6. 2017 - 2024 - تطورات

أصبحت JavaScript أساس تقنيات الويب الحديثة مثل React, Vue, Angular, Node.js.

تم تحسين الأداء وإضافة ميزات مثل `async/await`، `optional chaining`، والـ `private fields`.

تاريخ JavaScript يبدأ في عام 1995، عندما قام Brendan Eich، وهو مهندس في شركة Netscape، بتطويرها خلال 10 أيام فقط لتكون لغة برمجة موجهة نحو المتصفح. كان الهدف منها هو جعل صفحات الويب أكثر تفاعلية مقارنةً بـ HTML و CSS، اللتين كانتا تركزان فقط على المحتوى والتصميم. لماذا JavaScript مهمة اليوم؟

تعمل على جميع المتصفحات بدون الحاجة إلى تثبيت إضافي.

تُستخدم في تطوير تطبيقات الويب، الألعاب، تطبيقات الموبايل، وحتى الذكاء الاصطناعي.

تعتبر أكثر لغة برمجة شعبية في العالم وفقًا للإحصائيات الحديثة.

.....
Java script

Lesson one
.....

```
_a_
<!doctype html>
<Html>
<Head>
<Tital> in line java script
</tital>
</Head>
<Body>
<Hi>using in line java script
</h1>
<Script>
//Java script code here
Alert("helo world!");
<Script> helo,
```

```
<Body>
</Html>
```

```
.....
_b_
Let num =1.42;//nump
Let str="hello";//string
Let flag=true;//boolean
Let obj={key:"value"};object
Let arr=[1,2,3];//array
```

```
.....
_c_
If (condition){
//Code to execute if condition is true

Eals if (another condition){
//Code to execute if another condition is true
```

```
}Eals{
//Code to execute if no condition is true

}
```

```
.....
_d_
For( let i =0;i<5;i++){
//Code to repeat 5 times}
```

```
While (condition){
Code to repeat
While condition is true
}
```

```
.....
_e_
Function add (a,b){
Return a+b;

Let sum =add (2,3);//
Call the function
```

```
.....
```

```
_f_  
Let person={  
Name:"alice",  
Age:30,  
};  
Console.log (person.name)  
//Accessing object properties
```

.....
Lesson 2
.....

```
_a_  
<!doctype html>  
<Html>  
<Head>  
<Tital>in line java script</tital>  
<Script>  
//In line java script </tital>  
<Script>  
//In line java script goes here  
Function great (){  
Alert ("hello.world");  
}  
<Script>  
</Head>  
<Body>  
<Button onclick="great()">  
Click me </button>  
</Body>  
</Html>
```

.....

```
_b_  
<!doctype html>  
<Html>  
<Head>  
<Tital> async and defer </tital>  
<Script async scr="async-script.js">  
</Script>  
<Script deferscr="deferred-script .js">
```

```
</script>
</Head>
<Body>
<!--html content goes here-->
</Body>
<Html>
```

.....

Lesson 3

.....

```
_a_
<!doctype html>
<Html>
<Head>
<Tital>simple prompt and alert example
</Tital>
</Head>
<Body>
  <Script>
//Using prompt to get user input
Let user input =prompt
("Please enter your name:");
If(user input !==nully {
Alert (hello ," +user input+"!");
//Using alert to display a message
}Eals{
Alert("you didnt enter your name .");
}
</Script>
</Body>
<Html>
```

.....

```
_b_
<!doctype html>
<Html>
<Head>
<Tital>simple java script html
Out put </tital>
</Head>
```

```

<Body>
<H1 id="out but heading">java script
Html out but examble </h1>
<P id ="out put paragraph">
This is alert the java script out but will go.</p>
<Button on click ="change content()";
    Click me </button>
<Script>
Function change content (){
//Get references to html elements
-var heading element
=Document .get element byld("out but heading");
_var paragraph element
=Document .get element byld
("Out put paragraph");
//Change the content
Heading element .inner html
=New heading !";
Paragraph element .inner html=
"This content was added
Using java script.";
}
</Script>
</Body>
</Html>

```

.....
Lesson four
.....

a

```

//using var
Var x ;//declaration
X=10;//initialization

```

```

//Using let
Let y ;//declaration
    Y=20;//initialization

```

```

//Using const
Const z =30;//declaration and

```

initialization

```
Console.log(x);//out puts :10  
Console.log(y);//out puts :20  
Console log(z);//out puts:30
```

.....

b

Var global variable

=Im a global variable;

Function example function (){

Consle .log(global variable);//

//Accessing a global variable

}

Example function ();

Console.log (global variable);

//Accessing a globle variable

.....

c

If(true){

Let block var ="im block scoped variable";

Console .log (block var):

//Accessing a block _scoped variable }

//Console.log(block var);

„this would result in an error because block var is not defined here

.....

d

Console .log(x);//out puts:undifined

Var x =10;//variable declaration

Is hosted to the top

Function hosted function(){

Console .log("im hosted!");

.....

Lesson five

.....

a

If(true){

Let x

```
=10;
Console .log(x);output:10
}
Console .log(x);//error:
X is not defined
```

```
.....
_b_
Console .log(y);//reference
Error
Let y=20;
```

```
.....
_c_
Let counter=1;
Counter=2;reassignment is allowed
```

```
.....
_d_
Var global var =42;
Console .log(window.global var);
//42(in a browser)
Let global =42;
Console.log(window.global);
//Undefined(in a browser)
```

```
.....
Lesson six
```

```
.....
_a_
Const pi=3.14159
```

```
.....
_b_
If (true){
Const block scoped var=
"Im only available in this block";
}
Console.log(block scoped var:
//This will result in an error since block scoped var is not defined here
```

```
.....
_c_
Const pi=3.14159
Pi=42;//this will throw an error:assignment to constant
```


variable."

.....

```
_d_  
Const person={  
Name "alice"  
Age:30  
};  
Console.log (person.name);//  
Output:"alice"
```

You can modify the properties of the constant object

```
Person.name="bob"
```

```
//How ever,you cant reassign 'person'to a different object  
Person={name:charlie"}
```

```
//This will throw an error  
:assignment to constant variable."
```

.....

```
_e_  
Const numbers=[1,2,3,4,5]
```

```
Console.log (number[0];//output:1
```

```
//You can modify elements within the constant array
```

```
Number [0]=10;  
Console .log(numbers [0]);  
//Out puts :10
```

How ever ,You cannot reassign'numbers to a different array

```
Numbers[6,7,8,9]  
This will throw an error;  
"Assignment to constant variable."
```

.....

Lesson seven

.....

a

```
Let num=42;  
Let pi=14159;
```

.....

b

```
Let str="hello,world";
```

.....

c

```
Let is true=true;  
Let is false=false;
```

.....

d

```
Let un defined var;
```

.....

e

```
Let nul value=nul;
```

.....

f

```
Const unique symbol  
( 'description' )
```

.....

g

```
Let person={name:"alice,age:30};
```

.....

h

```
Let regex=/[0-9]+/;
```

.....

Lesson eight

.....

a

```
<!doctype html>  
<Html>  
<Head>  
<Tital>arithmetic operation  
  Example </tital>  
</Head>  
<Body>
```

```
<H2>arithmetic operation example
```

</h2>

<P id="result"></p>

<Script>

//Java script code starts here

//Arithmetic operations

Let num 1=10;

Let num 2=5;

//Addition

Let sum=num1+num2;

Document.get element byId

("Result").inner html+= addition :\${num1}+{num2}=\${sum}
;

//Subtraction

Let defference =num1-num2;

Document.get element byId("result")

.inner html+=subtraction :\${num1}×{num2}=\${prodect}
;

//Multiplication

Let prodect =num1×num2;

Document.get element byId

("Result").inner html +=multiplication

\${num1}×{num2}=\${prodect}

<Pr>;

Modulus

Let reminder =num%num 2;

Document.get element byId ("result")

.inner html +='devrsion:\${num1}/\${num2}=\${quation}

;

//Java script code ends here

</Script>

</Body>

</Html>

.....

Lesson nine

.....

a

```
Let temperature=25;
```

```
If(temperature>30){  
  Console.log("its not hot out side");  
}
```

.....

b

```
If (condition){  
  //Code to execute if the condition is true  
}Eals{  
  //Code to execute if the condition is false  
}
```

.....

c

```
<!doctype html>
```

```
<Html>
```

```
<Body>
```

```
<H2>java script if-eals</h2>
```

```
<Script>
```

```
Let age =15;
```

```
If (age>=18){  
  Document.write("you are an adult.");  
}Else{  
  Document.write ("you are a minor.");  
}
```

```
</Script
```

```
</Body>
```

```
</Html>
```

.....

d

```
<!doctype html>
```

```
<Html>
```

```
<Body>
```

```
<H2>hava script if ....else if...else  
</H2>
```

```
<Script>
```

```
Let grade=85;  
If(grade)>=90){
```

```
Document.write ("a");  
}Else if (grade z =70){  
Document .write("c")  
}Else{  
Document.write ("f");  
}
```

```
</Script>
```

```
</Body>
```

```
</Html>
```

.....

```
_e_
```

```
<!doctype html>
```

```
<Html>
```

```
<Body>
```

```
<H2>java script nested if-else </h2>
```

```
<Script>
```

```
Let x =10;
```

```
Let y =20;
```

```
If (x > 5){
```

```
If (y>10){
```

```
Document. write
```

```
(X is greater then 5 and y is greater then 10.");
```

```
}Else{
```

```
Document.write ("x is grater then 5,but y is not greater then 10.");
```

```
}  
</Script>  
</Body>  
</Html>
```

```
.....  
_f_  
<!doctype html>  
<Html>  
<Body>  
  
<H2>java script ternary operator </h2>
```

```
<Script>  
Let is runing =true;  
Let weather messege =is raining ?  
"Bring an emberlla";  
Document.write (weather messege);  
  
</Script>  
</Body>  
</Html>
```

```
.....  
Lesson ten  
.....  
<!document html>  
<Html>  
<Body>  
<H2>java script switch case</h2>  
<Script>
```

```
Let day=2;  
Let day name;  
  
Switch (day){  
Case one;  
Day name="sunday";  
Break;  
Case 2;  
Day
```

```
name="monday";
Break;
Case 3;
Day name ="tuesday";
Break;
Case 4:
Day name="wednesday"
Break;
Case 5;
Day name="thursday";
Break;
Case 6;
Day name friday";
Case 7:
Day name="saturday";
Break;
Default:
Day name="individual";
}
```

```
Document.write
("The day is "+day name);
</Script>
</Body>
</Html>
```

.....
Lesson eleven
.....

```
_a_
<!doctype html>
<Html>
<Body>

<H2>java script for loop</h2>

<Script>
For (let i=1;i<=5;i++)>{
Document.write(i+"<br>");
```

```
}  
</Script>
```

```
</Body>  
<Html>
```

```
.....  
_b_  
<!Doctype html>  
<Html>  
<Body>  
  
<H2>java script while loop</h2>
```

```
<Script>  
Let i=1;//initialization  
While (i<=5){ //condition  
  
Document.write(i+"<br>");  
i++;increment  
}  
</Script>  
</Body>  
</Html>
```

```
.....  
_c_  
<i Doctype html>  
<Html>  
<Body>  
<H2>java script do_while  
Lap</h2>
```

```
<Script>  
Let i=1;//initialization  
Do{  
Document.write(i+"<br>");  
i++;//increment  
}While{  
</Script>
```



```
</Body>
```

```
</Html>
```

```
.....
```

```
_d_
```

```
<!doctype html>
```

```
<Html>
```

```
<Body>
```

```
<H2>java script for-of loop
```

```
</H2>
```

```
<Script>
```

```
Const numbers=[1,2,3,4,5]
```

```
For(const num of numbers){
```

```
Document.write(num+"<br>");
```

```
}
```

```
</Script>
```

```
</Body>
```

```
</Html>
```

```
.....
```

```
_e_
```

```
<!doctype html>
```

```
<Html>
```

```
<Body>
```

```
<H2>java script for-of
```

```
Loop</h2>
```

```
<Script>
```

```
Const text ="hello".
```

```
For(const char of text){
```

```
Document.write(char+"<br>");
```

```
}
```

```
</Script>
```

```
<Body>
```

```
</Html>
```

```
.....
```

```
_f_
```

```
<!doctype html>
```

```
<Html>
```

```
<Body>
```

```
<H2>java script for-in loop</h2>
```

```
<Script>
```

```
Const person={  
Name:"alice";  
Age:30  
Profession:"engineer";  
};  
For(const key in person){  
Document.write(key+":"+person[key]+  
"<Br>");  
}  
</Script>  
</Body>  
</Html>
```

```
.....  
Lesson twilve
```

```
.....
```

```
_a_
```

```
For (let i=0;i<5;i++){  
//Some code here
```

```
If (condition){  
Break;//exit the loop if the condition is met  
}
```

```
//More code here  
}
```

```
.....
```

```
_b_
```

```
<!doctype html>  
<Html>  
<Body>
```

```
<H2>java script break </h2>
```

```
<Script>
For(let i=1;i<=5;i++){
If(i===3){
Break;//exit the loop when i is 3
}
Document.write(i+"<br>");
</Script>
```

```
</Html>
```

```
.....
_c_
For(let i=0;i<5;i++){
//Some code here

If (condition){
Continue;//skip the current iteration if the condition is met
}

//More code here
}
```

```
.....
_d_
<!doctype html>
<Html>
<Body>

<H2>java script continue</h2>
<Script>
For(let i=1;i<=5;i++)
{
If(i===3)}
Continue;//skip the iteration i is 3
}
Document.write<i+"<br>");
}
</Script>
</Body>
</Html>
```

.....
Lesson therteen
.....

a

```
Function add (a,b){  
    Return a+b;  
}
```

.....
b

```
<!doctype html>  
<Html>  
<Body>
```

```
<H2>java script function </h2>
```

```
<Script>  
Function say hello (){  
Document.write(hello, world!");  
}
```

```
//Calling the function
```

```
Say hello();
```

```
<Script>  
<Body>  
<Html>
```

.....
c

```
Function multiply(a+b){  
Return a*b;  
}
```

.....
d

```
<!doctype html>  
<Html>  
<Body>
```

```
<H2>java script function</h2>
```

```
<Script>  
Function
```

```
add(a,b){
Return a+b;
}
```

```
//Calling the function and storing the result in a variable
Let sum=add(5,3);
```

```
//Logging the result
Document.write("the sum is" :",sum);
</Script>
</Body>
</Html>
```

```
.....
_e_
Function function name(parameter,parameter);
//Function logic that uses parameters
}
```

```
.....
_f_
<!doctype html>
<Html>
<Body>
<H2>java script function with parameter and arguments </h2>
<Script>
Function add(a,b){
Return a+b;
}
Let result =add(5,3);
Document.write("the sum is :",result);
</Script>
</Body>
</Html>
```

```
.....
_g_
<!doctype html>
<Html>
<Body>

<H2>java script function with default
```

```
parameters</h2>
<Script>
Function greet(name="guest"){
Document.write("hello,"+name+"<br>");
}
```

```
Greet("alice");
Greet();
</Script>
</Body>
</Html>
```

.....
Lesson fourteen
.....

```
_a_
<!doctype html>
<Html>
<Body>
<H1>java script code method</h1>
<Script>
Function great(greeting){
Document.write(greeting+" "+this.name);
}
Const person={
Name:'john'
};
Greet.call(person,'hello');
</Script>
</Body>
</Html>
```

.....
Lesson fifteen
.....

```
_a_
<!doctype html>
<Html>
<Body>
```

```
<H2>java script apply method</h2>
<Script>
Function greet(greeting,punctuation){
Document.write(greeting+' '+this.name
+Punctuation;
}
Const person={
Name:'john'
};
Const args=['hello','!'];
```

```
Greet.apply(person.args);
```

```
<Script>
</Body>
</Html>
```

.....
Lesson sixteen
.....

```
_a_
</doctype html>
<Html>
<Body>
```

```
<H2>java script bind method</h2>
<Script>
Function greet(greeting){
Document.write(greeting+' '+this.name);
}
```

```
Const person={
Name:'john'
};
```

```
Const greet john=greet.bind
(Person,'hello');
Great john();
</Script>
```

```
</Body>
```

```
</Html>
```

```
.....  
Lesson seventeen
```

```
.....  
</doctype html>
```

```
<Html>
```

```
<Body>
```

```
<H2>java script bind method
```

```
</H2>
```

```
<Script>
```

```
Function greet (greeting){
```

```
Document.write(greeting+' '+ this.name>;
```

```
}
```

```
Const person={
```

```
Name:'john'
```

```
};
```

```
Const greet john =greet.bind
```

```
+Person, 'hello');
```

```
Greet john();
```

```
</Script>
```

```
</Body>
```

```
</Html>
```

```
.....  
_b_
```

```
Const person={
```

```
First name:"john",
```

```
Last name:"doe",
```

```
Age:30
```

```
Email:"johndoe@example.com
```

```
};
```

```
.....  
_c_
```

```
Console.log(person.first name);//
```

```
Console.log(person.age);//30
```

```
Console
```



```
log(person["email"]);"
```

```
.....  
_d_  
person.age=31;//modifying an existing property  
Person.city="new york";//  
Adding a new property
```

```
.....  
-lesson eighteen  
.....
```

```
_a_  
<!doctype html>  
<Html>  
<Body>  
  
<Script>  
Const person={  
First name:"osamah";  
Last name:"zuhair";  
Age:30  
Email:"osamahzuhair@example.com  
};
```

```
Document.write(person.firstname+"<br>");//"osamah"  
Document.write(person.age+"<br>");//30  
Document.write(person["email"]+"<br/>");
```

```
//"Osamahzuhair@example.com"
```

```
Person.age=31;//  
Modifying an existing property
```

```
Person.city="new york";  
//Adding a new property  
Document.write(person.age+"<br>"  
//"osamah"  
Document.write(person.city+"<br/>");30
```

```
</Body>
```

```
</Html>
```

```
.....
```

```
_b_
```

```
<!doctype html>
```

```
<Html>
```

```
<Body>
```

```
<Script>
```

```
//Define a constructor function for a person
```

```
Function(first name,last name,age){
```

```
This.first name=first name;
```

```
This.last name=last name;
```

```
This.age=age;
```

```
Method to get the full name
```

```
This.full name=function(){
```

```
Return first name+" "+last name;
```

```
Last name;
```

```
};
```

```
//Great two person object
```

```
Const person 1=new person
```

```
("Osamah","zuhair",30);
```

```
Const person 2=new person
```

```
("John","smith",25);
```

```
//Access properties and call method
```

```
Document.write(person1.full name()+"<br/>"
```

```
//Out put:osamah
```

```
Document.write(person2.age+"<br/>");
```

```
//Out put:25
```

```
Document.write(person1.full name()+"<br/>"
```

```
//Out put:osamah zuhair
```

```
Document.write (person2.full name()+
```

```
"<Br/>");
```

```
Out put:jone smith
```

```
</Script>
```

```
</Body>
```

```
</Html>
```

```
.....
```

```
_c_
```

```
<Doctype html>
```

```
<Html>
```

```
<Body>
```

```
<Script>
```

```
//Creating a boolean object
```

```
Var my boolean=new boolean(true);
```

```
//Getting the primitive boolean value
```

```
Var boolean value=my boolean value of
```

```
();//boolean value is true
```

```
Document.write("boolean value:"+boolean value+"<br/>");
```

```
</Script>
```

```
</Body>
```

```
</Html>
```

```
.....
```

```
Lesson ninten
```

```
.....
```

```
_a_
```

```
<!doctype html>
```

```
<Html>
```

```
<Body>
```

```
<Script>
```

```
//Great a constructor function
```

```
Function person(name,age){
```

```
This.name=name;
```

```
This.age=age;
```

```
}
```

```
//Add a method to the prototype of the constructor
```

```
Person.prototype.say hello=function()
```

```
Document.write("hello,my name is,this.name+"and im"+this.age+"years  
old.<br/>
```

```
};
```

```
//Creat instance of the person constructor
```

```
Const person 1=new person
("Alice",30);
Const person 2=new person
("Bob",25);
//Call the method defined of the prototype
Person 1.say hello();
//Out put:hello,my name is alice and im 30 years old.
Person 2.say hello();
//Out put:hello my name is bob and im 25 years old .
</Body>
</Html>
```

.....
Lesson twinty
.....

```
_a_
<!Doctype html>
<HTML>
<BODY>

<script>
Creating an empty set
Const my set=new set ();

//Creating a set from an array
Const my array=[1,2,3,4,5];
const my set from array=new set(my array);

Console.log(my set from array);
//Out put :set{1,2,3,4,5,}

</Script>
</Body>
</Html>
```

.....
b
<!doctype html>
<Html>

```
<Body>
```

```
<Script>
```

```
Const my set=new set();
```

```
My set.add(1);
```

```
My set.add(2);
```

```
My set.add(3);
```

```
Console.log(my set);//out put :set  
{1,2,3}
```

```
My set.delete (2);
```

```
Console.log(my set)://
```

```
Out put:set{1,3}
```

```
Console.log(myset.has(1));
```

```
//Output:true
```

```
Console.log(myset.has(2));
```

```
//Out put:false
```

```
</Script>
```

```
</Body>
```

```
</Html>
```

.....

```
_C_
```

```
<!doctype html>
```

```
<Html>
```

```
<Body>
```

```
<Script>
```

```
Const my set=new set[1,2,3];
```

```
//Using a for .....of loop
```

```
For(const item of my set){
```

```
Console.log(item):
```

```
}
```

```
//Converting to an array and using for each
```

```
Const my array=array.form
```

```
(My
```

```
set);  
My array.for each(item)=>
```

```
Console.loge(item);  
});
```

```
</Script>  
</Body>  
</Html>
```

```
.....  
_d_  
<!doctype.html>  
<Html>  
<Head>  
<Tital>java script sets example  
</Tital>  
</Head>  
<Body>  
<H2>fruit set</h2>  
<Bulid="fruit set"></ul>
```

```
<Script>  
//Creat a set  
Const a fruit set=new set();  
Fruit set.add('apple');  
Fruit set.add ('banana');  
Fruit set.asd('orange');  
  
//Get the ul element where we will display the list  
Const fruit list=  
Document.get element byld  
( 'fruit list);  
//Lterat over the set and add each item to the list  
Fruit set.for each(fruit)=>{  
Const list item=document -creat element ('li');  
List item.tect content=fruit;  
Fruit list.append child(list item);  
</Script>
```

```
</Body>
</Html>
```

.....
Leson twinty one
.....

```
(1)
<doctype html>
<Html>
<Body>
<Script>
Let num =3.14
Let formatted num=num.to fixed(2);
Doctype.write(formatted num.precision(4);
</Script>
</Body>
</Html>
```

.....

```
(2)
<!Doctype html>
<Html>
<Body>
<Script>
Let num=34;
Let numstr=num.toString();
Document.write(type of numstr+"<br>");
Document.write(numstr)
</Script>
</Body>
</Html>
```

.....

```
(3)
<!Doctype>
<Html>
<Body>
<Script>
Document.write(is num (24)+"<br>");
Document.write(is num(hello)))
</Script>
```

```
</Body>
</Html>
```

```
.....
(4)
<!doctype>
<Html>
<Body>
<Script>
Document.write(is finite(42)+"<br>");
</Script>
</Body>
</Html>
```

```
.....
(5)
<!Doctype html>
<Html>
<Body>
<Script>
Let pi=math.pi
Let rounded num=math.round(3.75)
Document.write(pi+"<br">));
Document.write(rounded num);
</Script>
</Body>
</Html>
```

```
.....
Lesson twinty tow
.....
```

```
(A)
<!doctype html>
<Html>
<Body>
<script>
//Creating an empty array using the array constructor
Let new arry =new arry();

//Creating an arry with initial values
Let fruits=new arry('apple,"banana";cherry)
```



```
</Script>
</Body>
</Html>
</Doctype html>
```

.....

(B)

```
<!doctype html>
<Html>
<Body>
<Script>
//Creating an empty array with specific length
Let new array =new array (5);
//Create an array with length of 5
</Script>
</Body>
</Html>
</Doctype html>
```

.....

(C)

```
<!doctype html>
<Html>
<Body>
<Script>
  Let fruits =['apple','banana','cherry'];
Document.write(fruits+"<br>");
Document write(fruits[0]+"<br>");
</Script>
</Body>
</Html>
```

.....

(D)

```
<!doctype html>
<Html>
<Body>
<Script>
Let fruits=['apple','banana','cherry'];
Fruit[1]='orange';
Document.write(fruits);
```

```
</Script>
</Body>
</Html>
```

```
.....
(E)
<!doctype html>
<Html>
<Body>
<Script>
Let fruits=['apple','banana','cherry'];
Document.write("iterate using for loop
:"+<br/>");
For (let i=0;i<fruits.length;i++){
Document.write(fruits[i]+<br/>")
Document.write("iterate using for each loop:"+<br/>");
Fruits.forEach(function(fruit){
Document.write(fruit+<br/>");
});
</Script>
</Body>
</Html>
```

```
.....
(F)
<!doctype html>
<Html>
<Body>
<Script>
//Creating an associative array using an object
Let person={
Name:"john";
Age:30,
City:"new york"
};
//Accessing values using keys
Document.write(person.name+<br/>");john
Document.write(person.age+<br/>");30
Document.write(person.city+<br/>");new york
//Modifying values
```

```
Person.age=31;
Document.write(person.age+"<br/>");31
//Adding new key_value pairs
Person.jop"software developer";
Document.write(person.jop+"<br/>");
//"Software developer"
//Deleting key-values pairs
Delet person.city;
Document.write(person.city);//undefined
</Script>
</Body>
</Html>
```

.....

Lesson twinty three

.....

(A)

```
#
Var is true=true;
Var is false=false;
Var is truthy="hello";//this is truthy
Var is false=0;//this is falsy
#
```

.....

(B)

```
#
Var x=true;
Var y=false;
Var result 1=x&&y;//logical and
Var result 2=x!!y//logical or
Var result 3=!x;//logical not
#
```

.....

(C)

```
#
Var a =5;
Var b=10;

Var is equal=a===b;//false
Var is greater then
```

```
=a>b;//false  
Var is less then or equal=a<=p;//true  
#
```

```
.....  
(D)  
#  
Var string value="true";  
Var boolean value=boolean(string value);  
True  
Var number value="0";  
Var boolean value 2=!!number value ://false  
#
```

```
.....  
(E)  
<!doctype html>  
<Html>  
<Body>  
<Script>  
Const true boolean object=new boolean  
(True);  
Const false boolean object=new boolean (false);  
Document.write(the boolean object);  
</Script>  
</Body>  
</Html>
```

```
.....  
(F)  
<!doctype html>  
<Html>  
<Body>  
<Script>  
Boolean.prototype.say hello=function(){  
Return "hello,im boolean object!";  
};  
Const my boolean=new boolean(true);  
Document.write(my boolean.say hello());  
</Script>  
</body>
```

</Html>

.....
(G)
<!doctype html>
<!Html>
<Body>
<Script>
Const true boolean=new boolean(true);
Const value =true boolean.value of ();

Document.write(type of vale+"
");
Document.write(value);
</Script>
</Body>
</Html>

.....
(H)
<!doctype html>
<Html>
<Body>
<Script>
Const true boolean=new boolean
(True);
Const str =true boolean string ();
//Returns "true"

Document.write(type of str+"
;
Document.write(str);
</Script>
</Body>
</Html>

.....
(I)
<!doctype html>
<Html>
<Body>
<Script>
Const a =true;
Const b=

```
true;
Const is equal=object is (a,b);
//Returns true
Document.write(type of is equal+"<br/>");
Document.write(is equal);
</Script>
</Body>
</Html>
```

.....

Lesson twenty four

.....

(A)

```
<!doctype html>
<Html>
<Body>
<Script>
Const current data=new data();
Document.write(current data);
</Script>
</Body>
</Html>
```

(B)

```
<!doctype html>
<!html>
<Body>
<Script>
Const specific code =new data
("2023-09-29ti0:30:00');
Document.write(specific data);
</Script>
</Body>
</Html>
```

(C)

```
<!Doctype html>
<Html>
<Body>
```

```

<Script>
Const data=new data();
Const year=data.get full year();
Const month=data get month();
Const day=date.get day();
Const hours=data.get hours();
Const minute=data.get minutes();
Const seconds=data.seconds();
Const milli seconds=data.get milli second();
Document.write("data:"+data+"<br/>");
Document.write("year:"+year+"<br/>");
Document .write("month:"month+"<br/>");
Document.write("day:"+day+"<br/>");
Document.write("minute:"+minutes+"<br/>");
Document.write("seconds:"+seconds+
"<br/>");
Document.write("milliseconds:"+
milliseconds);
</Script>
</Body>
</Html>

```

```

.....
(D)
<!doctype html>
<Html>
<Body>
<Script>
Const data=new data();
Data.set full year(2023);
Data.set month(8);september(0-based)
Data.set data(29);
Data.set hour (12);
Data set minutes(0);
Data.det second(0);

Decoment.write("date:"+date);
</Script>
</Body>

```

</Html>

.....

(E)

<!doctype>

<Html>

<Body>

<Script>

Const data=new data();

Const formatted data=data

.toLocaleDateString(); //formats as "mm/dd/yyyy"

Const formatted time=data

toLocaleTimeString(); //

formats as "h: mm:ss am/pm"

Const formatted date here

=Date.toLocaleDateString();

//Formats as "mm/dd/yyyy/hh:mm:ss

Am/pm"

Document.write("formatted data:"+formatted data+"
");

Document.write("formatted time:"+
");

</Script>

</Body>

</Html>

.....

(F)

<!doctype html>

<Html>

<Body>

<Script>

Const date =new Date();

Date.setDate (date.getDate()+7); //add 7 days

Date.setHours (date.getHours()-3);

//Subtract 3 hours

Document.write("modified date:"+date.getDate()+"
");

Document.write("modified hours:

date.getHours ()+"
");

</Script>

</Body>

</Html>

.....
Lesson twenty five

.....
(A)

```
<!doctype html>  
<Html>  
<Body>  
<Script>  
Document.write("math.pi:"math.pi+"<br/>");  
Document.write("math.E:"math.E+"<br/>");  
Document.write("math.ln2:"math.ln2+"<br/>");  
Document.write("math.ln10:"math.ln10+"<br/>");  
Document.write("math.log2E:"math.log2E+"<br/>");  
Document.write("math.log10E:"math.log 10 E+"<br/>");  
</Script>  
</Body>  
</Html>
```

.....
(B)

```
<!doctype html>  
<Html>  
<Body>  
<Script>  
Document.write("math.round(4.5):"+  
Math.round (4.5)+"<br>");  
Document.write("math.trance(4.9):  
+Math.trance(4.9)+"<br/>");  
Document.write("math.sign(5):"+math.  
Sign(5)+"<br/>");  
Document.write("math.sign(-5):"+math.sign(-5)+"<br>");  
Document.write("math.sign(0):"+math.  
Sign(0):"+math.sign(0)+"<br/>");  
</Script>  
</Body>  
</Html>  
</Doctype>
```

.....
.Lesson twenty six

.....

(A)

```
<doctype html>
<Html>
<Body>
<Script>
Class class name{
Constructor (parameters){
//Constructor function ,initialize object properties
}
Method(){
//Method definition
}
Method 2(){
//Another method definition
}
}
<Script>
<Body>
<Html
<Doctype html>
```

.....

(B)

```
<Doctype html>
<Html>
<Body>
<Script>
Class person{
Constructor (name,age){
This.name=name;
This.age=age;
}
Greet(){
Document.write
"(Hello,my name is"+this.name+"and im"+"this.age+"years old.<br/>");
}
}
//Creating instances of the person
```

```

class
Const person 1=new person("alice",30);
Const person 2=new person("bob",25);
//Calling the greet ();method on intance
Person 1.greet();//our put :hello
My name is alice and im 30 years old
Person 2.greet ();out put:
Hello my name is bob and im 25 years old
<script>
<Body>
<Html>

```

```

.....
(C)
<Doctype html>
<Html>
<Body>
<Script>
//Anonymous class experssion
Const circle=class {
Constructor(radius){
This.radius=radius;
}Area(){
Return math.pi*this.radius **2;
}
};
//Named class expression
Const square=class square shaip
Constructor (side length;

Area(){return this.side length **2;
Constcircle=new circle(5);
Const square=new square(4);

Document.write circle aria ()+<"pr/">>;
//Out but :78:539
Document.write(square.area());
//Output:16
</Script>

```

```
</Body>
```

```
</Html>
```

```
.....  
Lesson twenty seven  
.....
```

(A)

```
<doctype html>
```

```
<Html>
```

```
<Body>
```

```
<Script>
```

```
Class person{
```

```
Constructor(name,age){
```

```
This.name=name;
```

```
This.age=age;
```

```
}
```

```
}
```

```
Const person 1=new person("alice",30);
```

```
Const person 2=new person("bob",25);
```

```
Document.write(person1.name+"<br/>");
```

```
//Output:alice
```

```
Document.write(person1.age+"<br/>");
```

```
Document.write(person2.name+"<br/>");
```

```
//Output:bob
```

```
Document.write(person2.age+"<br/>");
```

```
//Output:25
```

```
</Script>
```

```
</Body>
```

```
</Html>
```

```
.....  
(B)
```

```
<!Doctype html>
```

```
<Html>
```

```
<Body>
```

```
<Script>
```

```
Class class name{
```

```
Constructor
```

```
(parameter1,parameter2){
}
Method name(){
//Method code
}
}
</Script>
</Body>
</Html>
```

```
.....
(C)
<!Doctype html>
<Html>
<Body>
<Script>
Class calculater{
Constructor(){
This.result=0;
}
//Method to add number
Add(a,b){
}
//Method to get the current result
Get result(){
Return this.result;
}
}
//Creat an instance of the calculator and class
Const my calculator=new calculator class
//Use the method to perform calculations
My colculator.add(5,3);
Document.write(my culcolator.get result()+"br/">);output=8
My calculator.subtract(2,4);
Document.write(my calculator.get result());
//Output:-2
</Script>
</Body>
</Html>
```

.....
(D)

```
<Doctype html>
<Html>
<Body>
<Script>
Class my class{
Constructor (param1,param2){
//Iniialize object properties here
This.property1=param1;
This.property2=param2;
}
}
</Script>
</Body>
</Html>
```

.....
(E)

```
<!doctype html>
<Html>
<Body>
<Script>
Class person {
Constructor(name,age){
This.name=name;
This.age=age;
}
}
//Creating instance using the constructor
Const person 1=new person ("alice";30);
Const person 2=new person('bob',25);

Document.write(person1.name+<"br/">);
Document.write(person1.age+<"br/">);

Document.write(person2.name+<"br/">);
Document.write(person2.age+<"br/">);
```

```
</Script>
</Body>
</Html>
```

.....

Lesson twinty eight

.....

(a)

```
<doctype html>
<Html>
<Body>
<Script>
Class math utilties{
//Static method to culcolate the cap of number
Static square (number){
Return number+number;
}
//Static method to culcolate the cup of number
Static cube (number){
Return number *number*number;
}
}
//Calling static method without creating instance
Const num1=5;
Const num2=3;

Const squired num1=
Math utilites .cub.(num);

Document.write ("square of "+num1+"squared num1+"<br/>");
Document.write ("cub of"+num2+" is" cubed num2);
</Script>
</Body>
</Html>
```

.....

(B)

```
<Doctype html>
<Html>
<Body>
```

```

<Script>
Class my class{
Static static property='im static property';
Static get static property;
}
}
Document.write (my class.static property+"<br/>");
Document.write(my class get static property+"<br/>");
//Accessing the static property via a static method
</Script>
</Body>
</Html>

```

.....
(C)

```

<Doctype html>
<Html>
<Body>
<Script>
Class math utilities{
Stac add(a,b){
Return a+b;
}
Static subtract (a,b){
Return a-b;
}
}
Const num1=10;
Const num2=5;

Const sum=math utilites.
Add (num1,num,);

Const difference=math utilities.
Subtract(num,num2);

Document.write("sum:"sum+"<br/>");
Document.write("defference"+difference);
</Script>

```



```
</Body>
```

```
</Html>
```

```
.....  
(D)
```

```
<Doctype html>
```

```
<Html>
```

```
<Body>
```

```
<Script>
```

```
Class math utilities {
```

```
Static add(a,b){
```

```
Return a+b;
```

```
}
```

```
Static add three number (a,b,c){
```

```
Return a,b,c;
```

```
}
```

```
}
```

```
Static add four num(a,b,c,d){
```

```
Return a+b+c+d;
```

```
}
```

```
}
```

```
Const num1=10;
```

```
Const num2=5;
```

```
Const num3=3;
```

```
Const num4=2;
```

```
Const sum1=math utilities.add
```

```
(Sum1,sum2);
```

```
Const sum2=math utilities.add
```

```
Three numbers(num1,num2,num3);
```

```
Const sum3=math utilities add four
```

```
Numbers(num,num2,num3,num4);
```

```
Document.write("sum1:"sum1+"<br/>");
```

```
Document.write("sum2:"sum2+"<br/>");
```

```
Document.write("sum3:"sum3+"<br/>");
```

```
</Script>
```

```
</Body>
```

```
</Html>
```

```
.....  
(E)
```

```
<Doctype html>
```

```
<Html>
```

```
<Body>
```

```
<Script>
```

```
Class my class
```

```
Constructor(){
```

```
//Call the static method with the constructor
```

```
My class.my class method();
```

```
Document.write('constructor executed.');
```

```
}
```

```
Static my static method(){
```

```
Document.write('static method calling.
```

```
"<Br/>");
```

```
}
```

```
}
```

```
Const instance=new class();
```

```
</Script>
```

```
</Body>
```

```
</Html>
```

```
.....  
Lesson twenty nine  
.....
```

```
(a)
```

```
<Doctype html>
```

```
<Html>
```

```
<Body>
```

```
<Script>
```

```
//Using closures to great variables
```

```
Function person(name){
```

```
Let privat name=(name);//privat variables
```

```
This.get name=function(){
```

```
Return private name;
```

```
};
```

```
}
```

```
Const person=new
```

```
person('alice');
Document.write(person.private name+"<br/>");
Document.write(person.get name());
//'alice'
</Script>
</Html>
</Body>
```

```
.....
(B)
<Doctype html>
<Html>
<Body>
<Script>
Function person(name){
This.name=name;
}
Person.prototype.sayhello=function(){
Document.write(hello,my name is"+"this.name+"<br/>");
};
Person1.say hello();//hello,my name is alice'
Person2.say hello();//hello, my name is bob
</Script>
</Body>
</Html>
```

```
.....
(c)
<Doctype html>
<Html>
<Body>
<Html>
Const person={
First name:'john',
Last name:'doe',
Get full name(){
Return this.{first name+" "+this.last name;
}
};
Document.write(person.full name);
//'john
```

```
doe'  
</Script>  
</Body>  
</Html>
```

```
.....  
(D)  
<Doctype html>  
<Body>  
<Html>  
<Script>  
Const person={  
First name:'john'  
Last name:'doe'  
Set full name(value){  
Const parts=value split(' ');  
The first name=part[0];  
The last name=part[1];  
}  
}  
Person.full name='alice smith';  
Document.write(person.first name+  
"<br/>");//'alice'  
Document.write(person.last name)://  
Smith  
</Script>  
</Body>  
</Html>
```

```
.....  
(E)  
<Doctype html>  
<Html>  
<Body>  
<Script>  
Class circle{  
This _radius=radius://private property with under score convention  
}  
Get raduis (){  
Return this . _radius;
```

```

}
Set radius (value){
If (value>=0){
This. _radius=value;
}Eats{
Document.write(radius)must be on - negetive. ');
}
}
}
Const circle=new circle(5);
Document.write(circle.radius+"<br/>");
Circle.radius=-2//radius must be non- negetive
</Script>
</Body>
</Html>

```

.....
Therty

```

.....
(a)
<doctype html>
<Html>
<Body>
<Script>
Class person
Construct(name){
This.name=name;
}
Say hello(){
Document.write(hello,my name is"+this.name+"<br/>");
}
}
Class student extends person{
Constructor(name,student id)
Super (name);
This.student id=student id;
}
Student(){
Document.write(this.name+"is studing.");
}

```

```
{
}
Const student 1=new student ("alice",12345);
Student 1.say hello();
//Out put:hello my name is alice'
Student1.study();
</Script>
</Body>
</Html>
```

.....

(B)

```
<Doctype html>
<Html>
<Body>
<Script>
Function person (name){
This.name=name;
}
Person.prototype.say hello=fuction(){
Document.write("hello,my name is"+this.name");
}:
Const person 1=new person("alice")
Person 1.say hello ();//out puts"hello,
My name is alice"
</Script>
</Body>
</Html>
```

.....

(C)

```
<Doctype html>
<Html>
<Body>
<Script>
Const person proto={
Say hello(){
Document.write("hello,my name is"
"+This.name);
};
```

```
};  
Const.person1=object creat(person proto);  
Person1.name="bob";  
Person1.say hello();//out put:  
"Hello my name is bob"  
</Script>  
</Body>  
</Html>
```

```
.....  
(D)  
<Doctype html>  
<Html>  
<Body>  
<Script>  
Function person(name,age){  
This.name=name;  
This.age=age;  
}  
Person.prototype.say hello=function(){  
Dicument.write("hello,my name is");  
};  
Const person 1=new person("john",30);  
Person1.say hello();  
</Script>  
</Body>  
</Html>
```

```
.....
```

