start java script learning 30/04/2022
* alert("hello world") //helo world
* console.log("hello world) //hello world
*document.write("hello world") //hello world
* single quation //
* dubble quation /* ..... */
* link javascript index.html
*Browser not support/not install javascript-- showing message(<noscript>.. </noscript>){
<noscript>Sorry, no JavaScript in this browser</noscript>
}
line end use ;
* "use strict"
* else if (a == "" || a == null || a == undefined) {
    alert("Try again")
}
* myDIV.
---
* text length chek exampall:
    let text = "Hello World!";
    let length = text.length; //12
* varible type 1. let name ; ------ decline
                  name = "hello" ; ----- define

---
* document.getElementById("age").style.color = "cyan";

* document.getElementById("age").innerHTML = result;

* document.getElementById("age").style.backgroundColor = "green";

* const xyz = document.getElementsByClassName("clee");
    xyz[0].innerHTML = "Hello World!";

* onclick = "muFunctions()"
    function muFunctions(){
        if(){

        }else{

        }
    }

* javascript display hidden none function{
    let hide = document.getElementById("hiden");

```
    if (hide.style.display == "none") {
        hide.style.display = "block";
    } else {
        hide.style.display = "none";
    }
}
---------

* varible 3 type 1. // type 1 :
                    let fname ;
                    fname = "foysal";
                    .....
                    let lname ;
                    lname = "ahmed" ;

                 2.   // type 2 :
                    let fname = "foysal" ;
                    let lname = "ahmed" ;

                3. // type 3
                    let fname = "foysal", lname = "ahmed" ;
* var === let
* var name use rolls  a-z, 0-9, $ _
            my_name, mu$naeme etc...
            1myname = wrong
* camelCess .. let myName = "Foysal" ;

* pascalCase .. let MyName = "Foysal" ;

* snakeCase .. let my_name = "Foysal" ;

-------------

* LEARN constant
* let variable can change
* const varible can't change
* const MY_SCHOOL_NAME = ("answer accedemy")
-----------
* datatype 1.string str= a-z
           2.integer int= 0-9
           3.floting = 34.34, .00

* java datatype is dynamic
* number type data 1.infinity ---25/0... [mean - big number of all the numbers ]
```

2.nun ---- hello/2 [mean - not a number]

* new type of value bigInt = 4543546765768780398495848229458n'use last = n' exm: const maxOfNumber = BigInt(65767645646543453); //65767645646543456n
console.log(maxOfNumber);

* 3 " " in line use "name\"Foysal" = let fas = "My name is Foysal\" i am a student" // My name is Foysla" am a student.

* sting data quation mark can use 1." " 2." 3. ``
@ quation can use `` 2 variable + exmpall: "cat"+price; == `cat  ${price} `; ----
```
const names = "Foysal Ahmed Shoun";
const age = 20;
const home = ' Shafipur\'s ';
const raselt = ` Hellow  ${names}  your age is  ${age}  you live's in ${home}`;
```

---

* Learning Boolean
* boolean data type is conditions True or False

---

* null datatype = empty value
* use null: let user = null ; user = "" ;
null = console.log("null")== object;
this is javascript team folt

---

* Undefiend datatype = user given no value

---

* which datatype of operator = how can I know, { let price = 100; } which datatypeof price , this is ..... typeof(price) == alert/console.log(typeof(price));
2 typeof = typeof price ; / typeof(price);

* typeof(name) //function form
* typeof name //operator form

*typeof convert number to sring use function {
```
let myNumber = "1234";
myNumber.toString();
```
}

==========

```
console.log(typeof(100)); //number
console.log(typeof("My Name")); //String
console.log(typeof(true)); //boolean
console.log(typeof(null)); //object
console.log(typeof(undefined)); //undefined
console.log(typeof(BigInt(56453654646345654))); //bigint
console.log(Object.keys(allworkers))
```

==========

---

* prompt == alert = console .log
* use prompt user can give input data exampall :

```
const pro = prompt("Enter you name","Foysla");
console.log(prom) //foysal
```

* prompt input placeholder data show use : const prom = prompt("Enter you name", "Foysal a ");
* history microsoft internet explorall old version not support prompt input placeholder,==undefined,, solved: prompt("Enter name"," ") ;

---

* Confirm = true/false == boolean (ok / cencle)

---

* Type of Conversion = number > type value change > string

```
string > type value change > number
exam:::
""const conversion = null //null
conversion = String(conversion) //string
alert (conversion) //string ""
 "" let conve = prompt("Enter a number", "10");
document.getElementById("c_con").innerHTML = typeof (conve);
conve = Number(conve);
document.getElementById("g_con").innerHTML = typeof conve; //Number ""
```

---

* Number Type of fail alert:

```
const checc = Number(undefined);
alert(typeof(checc)); //number
```

alert(checc); //NaN

* Number ("   23434   ") ; // 23434
* Number ("   3434a   ") ; // NaN
* Number ("   ") ; = 0
* Number (undefined) ; //NaN
* Number (true /'boolean type') ; // 1
* Number (false /'boolean type') ; // 0
* Number ("dskajfdlkajflkdj") ; // NaN
* Number (null) ; // 0

all link : https://dorey.github.io/JavaScript-Equality-Table/
---

* Boolean == empty value = False :
    nun / Undifined / NaN / 0 / "" / ---- "False"
       else: not empty value: "foysal" / 2 / 0 / (" ") --- True
                                          php (" ") --False

JavaScript operator

+ Addition x = y + 2
- Subtraction x = y - 2
* Multiplication x = y * 2
/ Division x = y / 2
% Modulus (division remainder) x = y % 2 : /end = vag ses
++ Increment x = y++
-- Decrement x = y--
** Exponentiation operator (**),= squar ;
const firstnumber = 5;
const number = 2;//5*5*5
=== 125 ans
let exm = 3**2*2 <== 18 (js read in backside)

* +, -, *, / == operator
* 1, 2, 3 = operand --- operator left - right =operand
* expration = (1+1) //full

=====part-2========

* Nigataion operator = -5
const firstnumber = 5;

```
        const secendnumber = -firstnumber;
        alert(secendnumber);
* Unary +operator = (same) :work with one number
                    let fn = Number("34");
                    let fn_2 = +("34")
                    alert(typeof fn_2);//number

* Binary Number: work  with two number = 5*9;
* Subcrabson opeartor = *

-----------------
* Concatenation :const fullname = fname + " " + mname +  " " + lname;

-----------------
* Operator Precedence: alert (1+1*9); // 10
* work with operator(* / + -) Priority
* if I do first + equal work use first braket ()= (1+1)*9;
* + - == same work with left;
* * / == same work with left;
* Operator Precedence power list link: https://www.w3schools.com/js/js_arithmetic.asp

--------------------
* Assignment Operators  ( = )
* let num = null;
alert(num = 4 + 9);
* chain Assignment
        let num;
        let num2;
        let num3;
       let num = num2 = num3 = 2;
        alert(num);
* object Assignment : +=, -=, *=, /=
        let number3 = 10 ;
        number3 = number3 + 30;
    short: number += 40 ;

---------------------
* Increment And Decrement Operators  ++, --
   let num = 1
   let num ++ -- // num = num + 1 ==2
* prefix = ++num     let noo = ++num // 2
* postfix = num++   let noo = num++ //1
* deffrence return value
* alert (num++) // 1
```

---

* Comma Operator :let com = (1 + 1, 2 + 2);
                  alert(com) // 4
* Comma Operator return last expration //4

---

* Comparison Operators : https://www.w3schools.com/js/js_comparisons.asp
1. == equal to -
2. === equal value and equal type
3. != not equal -
4. !== not equal value or not equal type
5. > greater than -
6. < less than -
7. >= greater than or equal to -
8. <= less than or equal to -

---

* Strict Equality Vs Equality / ekjak equal "===" "=="
  let se = "123" == 123; //true
  let se2 = "123" === 123 //false

---

* Null Undefined Equality
* let nue = null == undefined //true
* let nue = null == null //true
  null just given true use (null == undefined)
* let nue = null == 0 //false

==========part-3
---

* if statment
  if ( ){

  }
---

* else / else if statment
  if ( ){

```
    }else if(){

    }else{

    }
------------------

* Even Odd Program{
let numbers = Number(prompt("enter a number"));
numbers = Number(numbers);
if (numbers % 2 == 0) {
    document.getElementById("eoo").innerHTML = `${numbers} is a Even numbers`;
    document.body.style.backgroundColor = "orange";
} else {
    document.getElementById("eoo").innerHTML = `${numbers} is a Odd numbers`;
    document.body.style.backgroundColor = "gray";
}   }

--------------------

* Ternary Operator / Conditional /Question Mark Operator == if or else shortcut( ? :)
 ? : ? :
    {Expresson/ if  ?  if (resutl)  :  else(result);}
* Check Admin or Modaretor
* Exampal 1 {

    let userName = prompt("Enter your name:")
    let ter = userName == "Foysal" ? "You are admin" : "You are Modaretor";
    alert(ter);}

* Exampal 2{
    let test = +prompt("Enter your age");
    let result = test <= 17 ? `${test} You are baby boy` : `${test} Wellcome you are adult `;
    document.getElementById("age").innerHTML = result;
a   lert(result);
}
* Exampal 3{
    let pass = +prompt("number")
let text = pass >= 80 && pass <= 100 ? alert(pass + "You got a+") : pass <= 79 && pass >= 50 ?
alert(pass + "you got a") : pass >= 0 && pass <= 50 ? alert(pass + "you fail") : "";
}

----------------------
```

* logical operator ==
                    || == or
                    && == and
                    ! == not
                    ?? == null coalescing
* return boolean value

---

* || or operator need 2 condition to 1 condition true

* || or operator = || exam:== if(a>b || a>c) {

}

* || or operator find true valu

* or || operator true all value return = first value

* or || operator false all value return = last value

* or || operator can use value:
        let return = 0 || 12 || 40; // return true value 12

* when user cant give any value = else if (a == "" || a == null || a == undefined) {
  alert("Try again")
} ==== shortcut = let userName = prompt("Enter your name:");
  let nouserName = userName || "Blank user";
  if (nouserName == "Blank user") {
    alert("Plz input your name");
  } else {
    alert("Your name is: " + userName);
  }

---

* && == and operator

* && and opreator neet 2 condition true

* && and opreator find false value

* && == let return = 12 && 2 && 0 // return false value 0

* && operator true all value return = last value

* && operator false all value return = first value

! not operator
* ! not operator work !true == false , !false == true EXAM{
    let nameLenCheck = 1;
    if (!(nameLenCheck >= 3) && !(nameLenCheck <= 12)) {
        document.getElementById("nalen").innerHTML = "User Name Lenth Not Valid Use 3-12 lenth name"
    } else {
        document.getElementById("nalen").innerHTML = " User Name Lenth Valid";
    }
}
* !true == true , !false == true

?? Null Coalescing Operator ??

* ?? == (!== undefined and !== null)
* expresson or value ?? layout
* Ternary ternary shortcut exa:{
    let checks = "raj";

    let checkTer = (checks !== undefined) && (checks !== null) ? console.log("not nudefined") : console.log("undefined");

    == null coalescing;

    let nc;
    nc = checks ?? "undefined"
    console.log(nc); //raj
}

* ?? is new, old programmer use or || operator

* Difference ?? set = 0 return \\but or || give false

==========part-4

---all type of loop----

5 type of loop in javascript (while loop, do loop, for loop, for... in(select), for...of(all auto))
* Infinite Loop = again and again
* loop etaration = how much loop runing

---
* object loop for in{
    for in loop
}Exampal{
    let objectFor = {
    name: "Foysal",
    pation: "Student",
    age: 21
}
console.log(objectFor.name); // Foysal;

for (let key in objectFor) {
    console.log(key)
}
console.log(“Foysal” in objectFor)
////
for (let key in objectFor) {
    console.log(key, objectFor[key])
}

}
---
* arrays for of loop
* for of loop

* exampall-1{
    let arfas = ["Foysal", "Shoun", "Fardin"];

for (let fas of arfas) {
    console.log(fas) //Foysal", "Shoun", "Fardin"
}

}
----

    *while loop
* Structure {
    while (Condition){
        loop body

```
    }
}
* exampall:{
    let counter = 0;
    let con = "";
    while (counter < 5) {
        con += " <hr> try it:" + counter;
        counter++;
        counter--; // counter = 5
    }
    document.getElementById("wil").innerHTML = con;
}

-----------------------
    * do loop
* do loop work First time no condition true it's work , secend time condition must be true then
condition work.
* do loop Structure {

    do {
        loop body

    } while ( condition )
}
* exam{
        let doloop = 1;
        do {
        console.log("hello world " + doloop);
        doloop++;
    } while (doloop > 10);  //hello world 1
}

--------------
    for loop
* most use full loop

* for loop Structure {
    for ( begin ; condition checking ; steeping){
        loop body
    }
}

* exampall {
    for (let foLoop = 0; foLoop <= 10; foLoop++) {
```

```
    console.log("hello " + foLoop);
}
}
```

---

* Break Statement

* use loop stop

```
* exampall: {
    let num = 0;
    let numMas = "";
    while (num <= 10) {
        numMas += "<br> Break 10 of " + num;

        if (num == 5) {
            break;
        }
        num++;
    }
    document.getElementById("break").innerHTML = numMas;//5
}
```

* password same?
* prompt no close value add ,add

---

* Continue operatoe

* use loop Invisible value

```
* exampall {
    for (conti = 1; conti <= 10; conti++) {
    if (conti == 5 || conti == 6) {
        continue; // skip all thing
    }
    console.log("Hello " + conti);// 1 ,2 ,3 ,4, 7, 8, 9, 10
    }
}
```

========== part-5

---

* Nested Loop / inner loop

* exampall {
    user 1
 -----user 1
 -----user 2
 -----user 3
 user 2
 -----user 1
 -----user 2
 -----user 3
 user 3
 -----user 1
 -----user 2
 -----user 3
}

* nested loop console.log()

```
    let a;
    let c;
    for (a = 1; a <= 10; a++) {
        console.log("user " + a);
        for (c = 1; c <= 3; c++) {
            console.log("-----user " + c);
        }
    }
```

---

* lable break continue / break

* condition{

```
    lableName_1 :
    for( ; ; ){

        breakName_2 :
        for( ; ; ){
            break lableName_1 || breakName_2
        }
    }
}
```

* lable break use (break){

```
  l ableOne:
   for (x = 1; x <= 5; x++) {
      console.log("lableOne " + x);
      labelTow:
         for (y = 1; y <= 5; y++) {
            console.log("-------lableTow " + y);
            if (x == 2 && // || y == 2) {
               break lableOne;
            }
         }
   }
}
```

* lable break use (continue){

```
   lableOne:
   for (x = 1; x <= 5; x++) {
      console.log("lableOne " + x);
      labelTow:
         for (y = 1; y <= 5; y++)  {
            console.log("-------lableTow " + y);
            if (x == 2 && \\ || y == 2) {
               continue lableOne;
            }
         }
   }
}
```

---

* Switch Statement

* Switch Statement checking === strict operater

* Switch Statement use : when we need to work 1 value

*** switch statement every condition exit (case_- use " break" operator)

* switch statment body{

```
   let value = +prompt("100, 200", 100);
   if(value == 100){
      alert("Welcome You input " + value);
   else if (value == 200){
      alert("Welcome You input " + value);
```

```
    }else{
         alert(" You must input 100 or 200 You input " + value)
    }
    } Shortcut:-----

    let value = +prompt("100, 200", 100);
    switch (value) {
        case (100):
            alert("Welcome You input " + value);
            break
        case (200):
        case (300):
        case (400):
            alert("Welcome You input " + value);
            break
        default:
            alert(" You must input 100 or 200 You input " + value)
    }
}
```

---

* function call

* javascript not read function{}

* when funciton dcline - then js read function{}

```
* function myFunction() {
    console.log("hello");
    console.log("what");
    console.log("is");
}
myFunction();
myFunction();
```

---

* Function local variable & global variable

```
* local variable  assign in Function {
    let myName = " "  //local variable
}
```

* global variable assign out site .... let myName = " " //global variable

* local variable can use:- global variable

---

* Function parameters and Arguments

* which is fnction parameters and argument

```
   function pareArgu(fastName, lastName) \\Function Paraterers
   {
   console.log(fastName, lastName);
   }
pareArgu("Foysla", "Ahamed "); //Function arguments
```

* exampall 1{

```
   function funPera(hello, hello_2) {
   console.log((+hello) + (+hello_2));
   }
funPera(100, 300);
```

}

* exampall 2{

```
   function funPera(hello, hello_2) {
   console.log(hello = prompt("anything") + " " + (hello_2 = prompt("Enter a value")));
   }
funPera();
```

}

* exampall 3{

```
   function funPera(hello, hello_2) {
   console.log(hello + hello_2);
}
funPera(5, 10);

funPera(100, 399);
```

}

* arguments is a js key word

========== part-6

---

* Default parameter

* (Internet Explorer a old Harshany not work) function sum(namOne, namTow = "noValue")
//secend parameter == "no value" is Default parameter {
    console.log(namOne, namTow);
}
sum("fas") //secend parameter undefined

* exampall {
    function sumDafult() {
    alert("enter the secend parameter");
}

function sum(fName, lName = sumDafult()) {
    console.log(fName + " " + lName);
}
sum("Foysal")// no secend parameter
// enter the secend parameter

}
* default value we can set everything value, if condition (undefined ??), function, null colaecing (??), or(||) etc

---

* Function return exam{
    function fas(){
        alert("Hello World");
 //2. problem solved use(return("Hello World");   )
    }
    let fas_2 = fas() //1.return fas_2 variable undefined problem;
}

* function when use " return " then JavaScript exit the function -- like "loop" function use "break" statment;

* me ... user login use = id, and pass;{

    //   user id varify
const nam = prompt("Enter you name")

function userCheck(userId, userPass) {
    userCondition(userId, userPass);

```
    return userCondition(userId, userPass);
}

function userCondition(userId, userPass) {
    if (userId == 1234  && userPass == 1122) {
        document.getElementById("varefi").innerHTML = `Welcome ${nam}`;
    } else {
        document.getElementById("varefi").innerHTML = `Sorry ${nam} try again`;
    }
}
//button onclick;
function userFitback() {
    userCheck(prompt("userId"), prompt("Password"))
}
}
=====profesonal ====
//    user id Verify
function userCheck(userId, userPassword) {
    userCondition(userId, userPassword);
    return userCondition(userId, userPassword);

}
function userCondition(userId, userPassword) {
    const idPass = document.getElementById("varefi");
    // user - 1
    if (userId == "1234" && userPassword == "1122") {
        idPass.innerHTML = " Wellcome user1 "
        let userAccessCheck = true;
        return useAcc(userAccessCheck);
    } // user - 2
    else if (userId == "4321" && userPassword == "fas") {
        idPass.innerHTML = " Wellcome user2 ";
        let userAccessCheck = true;
        return useAcc(userAccessCheck);
    } // user - 3
    else if (userId == "1011" && userPassword == "use11") {
        idPass.innerHTML = " Wellcome user3 ";
        let userAccessCheck = true;
        return useAcc(userAccessCheck);
    } else {
        alert(" User id and password not match");
        idPass.innerHTML = "";
        let userAccessCheck = false;
        return useAcc(userAccessCheck);
```

```
    }
}
function useAcc(userAccessCheck) {
    if (userAccessCheck == true) {
        const acss = document.getElementById("userAccess");
        if (acss.style.display === "none") {
            acss.style.display = "block";
        }
    } else {
        document.getElementById("userAccess").style.display = "none";
    }
}
function userFitback() {
    userCheck(prompt("Enter your user id"), prompt("Enter your pass"))
}
```

---

* Javascript Prime Number : bangla(Moulik number : 1, 3, 5, 7, 9, 11 etc;)

* innerNumber start 2

* exampall{

```
// prime Number check
exiPrim:
    for (let outerNumber = 2; outerNumber <= 100; outerNumber++) {
        for (let innerNumber = 2; innerNumber < outerNumber; innerNumber++) {
            if (outerNumber % innerNumber == 0) {
                continue exiPrim;
            }
        }
        console.log(outerNumber); // 1  3 5 7
    }
}
```

* Dynamic exampall (use function parameter){

```
function primeNumberCheck(getPrimeNumber) {
    exirPrime: for (let firstNumber = 1; firstNumber <= getPrimeNumber; firstNumber++) {
        for (let secendNumber = 2; secendNumber < firstNumber; secendNumber++) {

            if (firstNumber % secendNumber == 0) {
                continue exirPrime;
```

```
            }
        }
        console.log(firstNumber);
    }
}
primeNumberCheck(50)
}

* use function get Prime Number exampall{

    //  Prime Number get Function
function getFstFun(userPrime) {
    let prin = ""
    for (let muNumber = 1; muNumber <= userPrime; muNumber++) {
        if (getSndFun(muNumber)) {
            let fontPrint = document.getElementById("funPri");
            fontPrint.innerHTML = prin += "<br> Use Functon Prime numebr is = " + muNumber;
            console.log(muNumber);
        } else {
            continue;
        }
    }
}

function getSndFun(muNumber) {
    for (let youNumber = 2; youNumber < muNumber; youNumber++) {
        if (muNumber % youNumber == 0) {
            return false;
        } else {
            return true;
        }
    }
}

function getFasFunCall() {
    let f = 20;
    getFstFun(f);
}
}

-------------------------
* type of funciton
* Function Expression / (Variable) use Function
```

```
* function variable(){
    return "Function use variable";
}

* (Variable) use Function rools >>

* let functionVariable = function (){
    return "Function use variable";
}; (*Must use)

 alert( functionVariable() ) //Function use variable

* factory Function
** factory Function{ // use camelCess;
    function myFunction(name, age, ) {
    let fas = {
        name: name,
        age: age,
        car() {
            console.log("Welcome to nasted function")
        }
    }
    return fas
}
let nam = myFunction();
nam.car(); //Welcome to nasted function
}

*  constractor function
** constractor function { //use pascalCase
    function constrac(name, age) {
    this.name = name,
    this.age = age
    this.car = function() {
        console.log("Welcome to nasted function");
    }
}

let prin = new constrac("Foysal", "Ahmed");
prin.car();
}

* class function{
    // declaring a class function
```

```
class product {
    constructor(item, price, discount) {
        this.item = item,
            this.price = price,
            this.discount = (
                price - (discount * price / 100)
            )
            //{
                finalDiscount() {
        return this.price - (this.discount * this.price / 100);
    } //bat.finalDiscount
            }
    }
}
let bat = new product("bat", 899, 10);

// Expresson a class function
let product1 = class { //product
    constructor(item, price, discount) {
        this.item = item,
            this.price = price,
            this.discount = (
                price - (discount * price / 100)
            )
    }
}
let boll = new product1(" boll", 1200, 15);
}

----------------------
*  extrand Function
exampall -1{
    // extrand product
class product_1 {
    constructor(items) {
        this.items = items;
    }
    product_Fun() {
        return this.items + " is a product";
    }
}
// extrand that
class Foysal extends product_1 {
    constructor(items) {
```

```
        super(items);
        //this.items2 = items;
    }
    product_Fun() {
        return this.items + " is a items";
    }
}
let pro1 = new product_1("Chair"); // pro1.product_Fun() = "Chair is a product"
let pro2 = new Foysal("Tabile"); // pro2.product_Fun() == 'Tabile is a items'

    }

exampall -2{
//extends person name
class mySelf {
    constructor(names, age, doing) {
        this.Name = names
        this.Age = age
        this.Pation = doing
    }
    person() {
        return this.Name + ' is a ' + this.Pation;
    }
}
let fas = new mySelf("Foysa", 22, "Diploma") // fas.person() = 'Foysa is a Diploma'

// class fardin extends{
class Fardin extends mySelf {
    constructor(names, age, doing) {
        super(names, age, doing)
    }
    person() {
        return this.Name + ' is a ' + this.Pation;
    }
}
let far = new Fardin("Fardin Ahmed", 16, "Student"); // far.person() == 'Fardin Ahmed is a
Student'

    }

----------------------------
* function gatter & satter mathod (get  set)=(change value on arrays use get and set value)
class product {
    constructor(name, price, discount) {
```

```
        this.name = name,
            this.price = price,
            this.discount = (
                price - (discount * price / 100)
            )
    }

    get getDiscoundt() {
        return this.discount;
    }
    set setDiscount(discount) {
        this.discount = discount
    }
}
let car = new product("car", 100, 5);
console.log( car.discount)// 5
console.log( car.getDiscoundt)// 5
console.log( car.setDiscount = 10)
console.log( car.discount)//10
```

---

* constractor function copy in varaible mathod{

```
(function cource() {
    this.title = "Java"
    this.enroll = function() {
        console.log("Welcome to enroll")
    }
})
// Replace
let cource = {
        title: "Java",
        enroll() {
            console.log("Welcome to enroll")
        }
    }
1. let functionCopy = new cource();
functionCopy.name = "Foysal"
console.log(functionCopy) //cource {title: 'Java', name: 'Foysal', enroll: ƒ}.

2. // Object.assing
const functionCopy = Object.assign({}, cource);
functionCopy.name = "Foysal";
```

```
console.log(functionCopy)

3.  // loop in
for (let key in cource) {
    console.log(key, cource[key])
}

4.  // loop of
for (let key of Object.keys(cource)) {
    console.log(key, cource[key])
}

5. // loop of copy value
const functionCopy = {};
for (let key of Object.keys(cource)) {
    functionCopy[key] = cource[key];
}}
====================== end: video
* java script object

 ----------------------------
 * java script Trim

 * exampall 1{
     let text = "     Hello World!     ";
let result = text.trim();

document.getElementById("demo1").innerHTML = text;  //      Hello World      .
document.getElementById("demo2").innerHTML = result;//Hello World.
 }

 * exampall 2{
     let userValue = prompt("Enter your user name");
let userValueCheckTrim = userValue.trim();
let userValueCheck = userValueCheckTrim || "blank";

let ter = userValueCheck == "blank" ? " Plz input you name " : "";

if (userValueCheck == userValue) {
    alert(`Welcome ${userValue}`);
} else {
    console.log(ter);
}
 }
```

---
* Dom === The Document Object Model

---

* Object in javascript

1. variableName.Object;
2. variableName["Object"]; // value

Exampall-1{
    const car = {
    brand: "Farire",
    color: "Red",
    id: 789
};
console.log(car.brand);//Farire
}

exampall-2{
    const car = {
    brand: "Farire",
    color: "Red",
    id: 789
};
console.log(car["id"]);
}

---
* string differnce type of Methot

    let yourName = " Foysla ";
* length methot:
    * string.length;
    let yourNameLen = yourName.length;
    alert(yourNameLen) // 6

* string value :
     yourName[2];
    console.log(yourName[2]); // y

* string value number :
    yourName.indexOf("h");

```
    console.log( yourName.indexOf("h") ) //8

* String full sentence to get number :
    yourName.slice(7, 11)
     console.log(yourName.slice(7, 11)) //ahme

* String upperCase BigLatter :
    yourName.toUpperCase();
    console.log(yourName.toUpperCase()); // FOYSAL AHMED SHOUN

* String lowerCase smallLatter :
    yourName.toLowerCase()
    console.log(yourName.toLowerCase()) // foysal ahmed shoun

------------------

* querySelector:
    * let htmldoc = document.querySelector(".user_input").style.backgroundColor = ("red");

* querySelectorAll:
    let sam = document.querySelectorAll("p");
    for (let fas = 0; fas <= sam.length; fas++) {
        sam[fas].style.color = ("red");
        console.log("color " + fas)
    }


------------------
* createElement

    const btn = document.createElement("button");
    btn.innerHTML = "Hello Button";
    document.body.appendChild(btn);

//////////////----------

* Problem solve - 1:
     // first alphabet Big and last all small

    let names = ["foYsal", "fArdin ", "swaroN", "sisHir", "sHuvo"];
    const sam = document.getElementById("upper");
    let samCheck = "";
    for (let i = 0; i <= names.length; i++) {
        let lowerCase = names[i].toLowerCase();
```

```
        let firstSlice = lowerCase.slice(0, 1);
        lowerCase = lowerCase.replace(firstSlice, firstSlice.toUpperCase());

        samCheck += "<br>" + lowerCase;
        sam.innerHTML = samCheck;  //Foysal  Fardin  Swaron  Sishir Shuvo
    }

* Problem solve - 2:
// text Search input

    let storys = [
        " in th World",
        " hello world", " welcome in the ", " welcome in hello world", " yes hello"
    ]
    let not = document.querySelector(" ul li");
    //console.log(storys.length);
    //console.log(storys.indexOf("World"))
    for (let tex = 0; tex <= storys.length; tex++) {

        if (storys.indexOf("hello") !== -1) {
            console.log(storys);
            let com = document.createElement("h1");
            com.textContent = (storys);
            not.appendChild(com);
        } else {
            console.log(1)
        }
    }

* Problem solve - 3:
  // get Elements / vs / quarySelector && createElement && textContent && appendChild

    let test = " ldkjfsldfjlds kdjflksjdlfkeowj ij ksfsj lk23 3943094 kjflkdsjflkjdfljfdsljf kjlkjdfj lkfjdksj fjla
"

    // use get Elements

    let a = document.getElementsByClassName("background");
    a[0].innerHTML = test;

    // quarySelector && createElement && textContent && appendChild

    let b = document.querySelector(".background");
    let c = document.createElement("h6");
```

```
    c.textContent = test;
    b.appendChild(c);

---------------
* make calculetor{
    <div class="calculetor">
        <form name="form">
            <div class="display">
                <input id="displayed" type="text" name="answer">
            </div>
            <div class="oneTowinput">
                <input type="button" value="1" onclick="form.answer.value += '1' ">
                <input type="button" value="2" onclick="form.answer.value += '2' ">
                <input type="button" value="3" onclick="form.answer.value += '3' ">
                <input type="button" value="4" onclick="form.answer.value += '4' ">
                <input type="button" value="5" onclick="form.answer.value += '5' ">
                <input type="button" value="6" onclick="form.answer.value += '6' ">
                <input type="button" value="7" onclick="form.answer.value += '7' ">
                <input type="button" value="8" onclick="form.answer.value += '8' ">
                <input type="button" value="9" onclick="form.answer.value += '9' ">
                <input type="button" value="0" onclick="form.answer.value += '0' " id="gro">
            </div>
            <div class="aditional">
                <input type="button" value="+" onclick="form.answer.value+= '+' ">
                <input type="button" value="-" onclick="form.answer.value+= '-' ">
                <input type="button" value="*" onclick="form.answer.value+= '*' ">
                <input type="button" value="/" onclick="form.answer.value+= '/' ">
                <input type="button" value="=" onclick="form.answer.value= eval(form.answer.value)
">
                <input type="button" value="clean" onclick="form.answer.value = '' ">
            </div>
        </form>
    </div>
}

----------------------
*Javascript : Top  Websites to practice | learn resource:
* video link : https://www.youtube.com/watch?v=vldRjjbcRXc&list=LL&index=1
1. https://www.jshero.net/en/home.html
2. https://edabit.com/

------------------
* Remove / delete {
    var person = {
```

```
  firstName:"John",
  lastName:"Doe",
  age:50,
  eyeColor:"blue"
};

delete person.age;  // or delete person["age"];
}
```

```
* Type of Event( check tutrial point js EVENT-14 page:74){
    1.onclick="sayHello()" ;

    2.onsubmit Event Type , (The following example shows how to use onsubmit. Here we are
calling a validate() function before submitting a form data to the webserver. If validate() function
returns true, the form will be submitted, otherwise it will not submit the data.)

    3.onmouseover && onmouseout(
        onmouseover="this.style.color='blue'"
        onmouseout="this.style.color='orange'"
    )
    * HTML 5 Standard Events(Offline,ondrag): pageno: 78
}
```

```
* Everything about learn arrays

1. print arrays mathid{
    let num2 = [1, 2, 3, 4, 5, 6, 7, 8, 9];
console.log(num2);
document.write(num2)
    }
// tatal array last value
* console.log(num2[num2.length - 1]);// last arrays = 9
2. print arrays mathid{
let num2 = new Array(1, 2, 3, 4, 5, 6, 7, 8, 9);
console.log(num2);
    }
** add arrays remove arrays{
    let num2 = new Array(1, 2, 3, 4, 5, 6, 7, 8, 9);
// push add a value on array in last
num2.push(10) // [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
    // unshift add a value on array in fast
num2.unshift(0) //[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
    // Removes the last element
num2.pop() // [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
    // Removes the fast element
num2.shift() // [1, 2, 3, 4, 5, 6, 7, 8, 9]
console.log(num2);
    }
3 array element change. num2[0] = "start"; // ['start', 2, 3, 4, 5, 6, 7, 8, 9]

* Array indexOf{
let everyName = ["Faysal", "Fardin", "Swaron", "Shuvo", "Fardin", "Shishir"];
document.write("<br>" + everyName.indexOf("Fardin")) // 1

-- 2 time same element in array

let everyName = ["Faysal", "Fardin", "Swaron", "Shuvo", "Fardin", "Shishir"];
document.write("<br>" + everyName.indexOf("Fardin", 2)) // 4

 * last indexOf = console.log(everyName.lastIndexOf("Fardin")) //4

 *same value match includes return true, false
console.log(everyName.includes("Swaron")) // true

console.log(everyName.includes("Swaron", 3)) // false

** .find array in Object Search {
    let youTube = [{
        name: "Apna Collage",
        subscriber: 10000
    }, {
        name: "Fas world",
        subscriber: 20000
    }

]
1.let search = youTube.find(function(fass) {
    return fass.name === "Apna Collage";
})
console.log(search) //name: 'Apna Collage', subscriber: 10000}
    }
    }
2.let fin = story.find(element => element.Channel === "Hello")

** add array use Concatenation
let names = ["Foysal", "Fardin", "Shoun"];
```

```
let names1 = ["raka", "Fahim", "Himel"];

console.log(names.concat(names1))

---same mathod use spread
let names3 = [...names, ...names1]
console.log(names3)

* Remove array value
let nam4 = names3.slice(3, 5)

-- find array value use loop

1. for (let i = 0; i < names.length; i++) {
    console.log(names[i])
}
2. for (let key of names) {
    console.log(key)
}

* join a string
let joinString = ["F", "o", "y", "s", 'a', 'l']
console.log(joinString.join("-"))//F-o-y-s-a-l
* split {
    let joinString = ['F', 'o', 'y', 's', 'a', 'l'];
let student = joinString.join("_");
console.log(student.split("_"))
}
-- filter (same as find but batter then find(Get all > value))
1. let countryInformation = [{
    City: "Dhaka",
    Population: 20000
}, {
    city: "Gazipur",
    Population: 5000
}, {
    city: "Kaliakoir",
    Population: 3000
}]
let findCoutyInfo = countryInformation.filter(citys => citys.Population > 4000)
console.log(findCoutyInfo)

--- map (Select All value array)
1.let findCoutyInfo = countryInformation.map
```

```
(citys => citys.Population * 2)
console.log(findCoutyInfo)
2. (Exa) let nameHeight = character.map(fas => {
    return {
        Name: fas.Name,
        Heigth: fas.Height
    }
})
--- total number .reduce(){
    1. let totalNumber = [1, 4, 5, 10];
let getTotalNum = totalNumber.reduce((previus, newValue) => {
    return previus + newValue; //20
})

    2. let getTotalNumber = character.reduce((previous, newPlas) => {
    return previous + newPlas.Height
    }, 0);  //16.4

}

--------- spread vs r rest opeartor------
// function Parameter is this rest operator
        let car = ['toyata', 'honda', 'farrery', 'tesla'];
        allCars = (...allCar) => {
            return (allCar)
        }
        // function puss value is this spread operator
        let printCar = allCars(...car);
        console.log(printCar);

-------- ganerator function * click button step by step -----
<button onclick="stepByStep()">Click an open console</button>
        function* steps() {
            console.log('1st step loading');
            yield '1st step compleate';

            console.log('2st step loading');
            yield '2st step compleate';

            console.log('3st step loading');
            yield '3st step compleate';
        }
        let startStep = steps();
```

```
        function stepByStep() {
            console.log(startStep.next())
        }

----------------------
----------DOM dom Dom-----------
* window.onLoad()
* location.reload()
* localStorage.setItem(p1,p2)
* localStorage.key(1)
* localStorage.get(p1)
* localStorage exampall { https://onecompiler.com/html/3ydmtna2c }
* audio file add exam:{ let tap = new Audio('../sound/tap.mp3')}
1. console.log(document.body.documentElement)
2. console.log(document.body.childNodes)
3. console.log(document.body.firstElementChild)
4. console.log(document.body.lastElementChild)
5. sibling  // nextElementSibling  // previousElementSibling // textContent
const sibling = (document.body.children[1]);
console.log(sibling);
console.log(sibling.nextElementSibling);
console.log(sibling.previousElementSibling.textContent);
console.log(currentElement.parentElement);
6.
// find childrens childern and change backgroundColor
table>tr*3>td*6(caption)
// use loop
for (let a = 0; a <= 5; a++) {
    document.body.children[0].tBodies[0].rows[0].cells[a].style.backgroundColor = 'green';
}

// 1st
const tableTag1 = document.body.children[0];
tableTag1.tBodies[1].rows[0].cells[0].style.backgroundColor = 'red';
// 3rd
tableTag1.tBodies[1].rows[0].cells[2].style.backgroundColor = 'green';
// 4th
tableTag1.children[1].rows[0].cells[4].style.backgroundColor = 'yellow';
// 5th
tableTag1.children[1].rows[1].cells[1].style.backgroundColor = 'white';
// 6ht
tableTag1.children[1].rows[1].cells[3].style.backgroundColor = 'purple';
// 7ht
tableTag1.children[1].rows[1].cells[5].style.backgroundColor = 'pink';
```

```
// 8ht
tableTag1.children[1].rows[2].cells[0].style.backgroundColor = 'cyan';
// 9ht
tableTag1.children[1].rows[2].cells[2].style.backgroundColor = 'blue';
// 10th
tableTag1.children[1].rows[2].cells[4].style.backgroundColor = 'brown';

// queary selector fast child & last child select
1. const quSelect = document.querySelector('ul>li:last-child')
console.log(quSelect.textContent) //Third

2.const quSelect = document.querySelector('ul>li:nth-child(2)')
console.log(quSelect.textContent) //Secend

-----------------
// Access predefine html (id) or (counsom) id
     <table id="tab" data="valid">
//predefine html (id)
1.let oc = tab.id;  // tab

2.let oc2 = tab.getAttribute('data');  // valid

3.// setAttribute
tab.setAttribute('new', 'new-find') // attribute name , attribute value
let oc2 = tab.getAttribute("new")  // new-find

4.// hasAttribute (attribute exist or not)
let oc2 = tab.hasAttribute("new") // true || false

5.// removeAttribute
tab.removeAttribute('new')

6. //get all attributes
let oc2 = tab.attributes

-----------------------
1. link http requirement = let select = `a[href*='://']`
2. specific link requir = let select = `a[href*='://']:not([href="http://internal.com/test"])`;
 {let qsel = document.querySelectorAll(select);
   qsel.forEach(element => element.style.backgroundColor = 'red');}
----------------------
like append - Child
1. document.body.append(creat); // buttom append
2. document.body.prepend(creat); // top append
```

3. document.body.before(creat); // top in body
4. document.body.after(creat); // buttom in body

---
replaceWith
1. id tag bady = body.replaceWith(story)
2. replace element / replace child = currElement.parentElement.replaceChild(newElement, replaceElemnt)
---
delete forever id
function deleteFunction() {
    let namDelet = document.getElementById('delet');
    namDelet.remove();
    console.log(namDelet)
}

----------events----------
function callMe() {
    alert('What is this, Why you calling me like that');
}
1. onclick
(idName) ClickMe.onclick = callMe;

2. addEventListener
clickMe.addEventListener('click', callMe);
document.addEventListener('click', callMe);

3. removeEventListener
clickMe.removeEventListener = ('click', callMe); //.....

4. Click event courser X Y access , typeof button or else, events type
function callMe(evnet) {
    console.log('What is this, Why you calling me like that'); // What is this, Why you calling me like that
    console.log(evnet.type); // click
    console.log(evnet.currentTarget); // <input type="button" name id="clickMe" value="Click Me Plz">
    console.log(evnet.clientX); // 65
    console.log(evnet.clientY); // 308
    console.log(evnet.currentTarget); // select input value .courrentTarget || previousElementSibling select

}
clickMe.addEventListener('click', callMe);

-----make clock auto secend count----------{
1. let date = new Date();
   let hh = date.getHours();
   let mm = date.getMinutes();
   let ss = date.getSeconds();
   let session = 'am';

}
2. //function key = setTimeout(function() {
            main(2)
        }, 2000);
3. if (hh > 12) {
      hh = hh - 12;
      session = 'pm';
   } else if (hh == 0) {
      hh = hh + 12;
   }
----- class add class new create element----------
1. classList.add('className');
2. className = 'className';
—------ timeOut —--
   1. setTimeOut(()=>{ functionName() },2000) ||
      setTimeOut(functionName,time(2000),”paramiter”’)
   2. setInterval – auto call
   3. clearTimeout(var)
--- css animation ---
1.{
   .animate {
   animation: animatetop 0.4s
}

@keyframes animatetop {
   from {
      opacity: 0
   }
   to {
      opacity: 1
   }
 }
2. opacity 0-1
3. Position static(dafult) absolute, relative, fixed and *(scrool)sticky in CSS
}
4. .class{

```
    dafultdisplay: grid;
    grid-template-rows: repeat(18, 1fr);
    grid-template-columns: repeat(20, 1fr);
    }
------css responsive------
1. em, rem, vh and vw units + Responsive design
2. @media only screen and (max-width:955px)
3 @media only screen and (min-width:955px)

------- option select value get---------
1. let select = document.getElementById('cars');
console.log(select.options[select.selectedIndex].value)
----window refresh----
1. function refreshPage() {
    window.location.reload();
}

----- keybord top down key----
1.{
    window.addEventListener("keydown", f => {
    exes = {
        x: 0,
        y: 1f
    }
    move.play()

    switch (f.key) {
        case ('ArrowUp'):
            console.log('up');
            break;
        case ('ArrowLeft'):
            console.log('left');
            break;
        case ('ArrowRight'):
            console.log('Right');
            break;
        case ('ArrowDown'):
            console.log('down');
            break
    }
}

----- create / ganerate new number / item -----
1.{
```

```
   foodOb={
      x :1,
      y: 5
   }
   let a = 0;
      let b = 16;
      foodOb = { x: Math.round(a + (b - a) * Math.random()), y: Math.round(a + (b - a) *
Math.random) }
}
```

—----requestAnimationFrame—--

St-1: `window.requestAnimationFrame(person)`

St-2:

```
function person(pel1) {
          window.requestAnimationFrame(person)
      }
```

----------Use api-------

```
function appiFun(){
   const url = "https://api.covidtracking.com/v1/us/daily.json";
   fetch(url).then(response => response.json()).then(data=> {
      data.forEach((value) => {
       latitude = value.latitute;
       longitude = value.longitude;
       // my
      })
   })
}
appiFun()
```

—---- favourite extensions—--------

1. Js css html formatter

—-------- learn React js —---------

1. Install node js
2. node –version
3. npm –v
4. npx –v
5. Install react js command { npx create-react-app folderName react }
6. npm start
7. Chack all folders command = ls

8. Go under the folder command = cd folderName
* . back folder = cd..
*. Which folder = pwd
*. Delete file = rm -file name
*. edit file = vim (file name) => i (Insert) => after edit press (ESC) key => save and exit (shift + (:) wq + enter)
*. Sudo nano / dpkg (file name)
*. Sudo apt install (pakage name)
*. Sudo su - (for root user)
*. Sudo nautilus == for access add document
9. App.js under app() function add code problem solved =
<React.Fragment>...</React.Fragment>
10. Add new file in index.html/Formate Document = command + shift + p
11. Convert html to jsx link: https://transform.tools/html-to-jsx
12. Make class component on React.Component cc / rcc / rfc
13. Creat new component must import == imr / import React from “react”;
14. Pass data in component use function (props)
15. With out install react js link {
<script src="https://unpkg.com/react@16/umd/react.production.min.js"></script>
<script src="https://unpkg.com/react-dom@16/umd/react-dom.production.min.js"></script>
<script src="https://unpkg.com/babel-standalone@6.15.0/babel.min.js"></script>
}

—-----Learn Node js node.js—----

- Node module

1 . os , path, file system
fs – read file exm{

```
fs.readFile('name.js', 'utf-8', (err, data) => {
      console.log(err,data);
});
```

}
fs - read file Sync exm{

```
// readFileSync
let x = fs.readFileSync("name.js");
console.log(x.toString())
```

}

2. CJS vs EJS **CommonJS Modules** (CJS) and EcmaScript Modules (ESM)
3. Install npm fame work Nodemon, Mongoose express js, express pug, postman || dogApi, jsonPlaceholder
4. Create local host Server exm{

```
const http = require('http');
const fs = require('fs');
const hostname = '127.0.0.1';
const port = 3000;
const server = http.createServer((req, res) => {
   console.log(req.url)
   res.statusCode = 200;
   res.setHeader = ('Content-type', 'text/html')
   if (req.url == '/') {
       res.end('<h1>Welcome</h1>')
   } else if (req.url == '/home') {
       res.end('<h1>Welcome home</h1>')
   } else if (req.url == '/file-1') {
       const file = fs.readFileSync('index.html');
       res.end(file.toString())
   } else {
       res.end('Url error')
   }
})
server.listen(port, hostname, () => {
   console.log(`Server running at http://${hostname}:${port}/`)
}) OR
app.listen(port,() => {
   console.log(`Server running at localhost:${port}/`)
})
```

}

5. * Host buy server DigitalOcean.
6. * Learn nodejs website docs
7. Learn Express full courch
8. Software version => 1.2.5 = M.M.P == Mazor(mazor fun change) . M(small fun add) . Patch(bug fiex);
9. `path.join(__dirname, 'src').`
10. nam (p.name) –save -dev ==(devDependencies)

—------ node js express —-----

1. Get from data form data{

```
// Require file
const express = require('express');
const fs = require('fs');
const path = require('path');
const app = express();
// set express engine
—--npm install pug
—-app.set('view engine', 'pug')

// use include file for image
 app.use('/static', express.static('../static'))
// use path // render page
app.set('views', path.join(__dirname, 'src'))
app.get('/dog', (req, res) => {
   res.render('demo', { title: 'Hey', message: 'Hello there!'
})
})
// get-post data on form
app.use(express.urlencoded())
app.post('/dog', (req, res) => {
   fName = req.body.fName;
   lName = req.body.lName;
   gmail = req.body.gmail;
   password = req.body.pass;
   more = req.body.tArea;
let userDataInputFile = `---The user First name= ${fName}, last
name= ${lName}, gmail= ${gmail}, password= ${password}, about
yourself ${more}`;
fs.writeFileSync('../user.txt',userDataInputFile);
res.status(200).render('demo', {  "content": cont });
}
app.listen(port,() => {
   console.log(`Server running at localhost:${port}/`)
})}
```

#. Send data mongodb install => npm install body-parse

—------ Mongo db—------

C.R.U.D => Create, read, update and delete

COMMANDS basic

1. Start == mongod mongosh
2. == show dbs
3.== use foysal = switched to db foysal
14.== show collections

Create

4== db.text.insert({"name":"Foysal"})

6 ==

```
db.items.insertOne({name:"xioami",price:12000,kuntiti:120,sell:99,rating:4.5})
```

```
db.intems.insertMany([{name:"xioami",price:12000,kuntiti:120,sell:99,rating:4.5},{name:"iphone",price:112000,kuntiti:90,sell:80,rating:4.9}])
```

read

db.intems.find({rating:4.5})

7== db.items.find() , db.intems.find({rating:3.5}) , >= db.intems.find({rating:{$gt: 3.5}}) , == gt , < lt

update

== db.items.updateOne({name:"xioami"},{$set:{price:2}})

== db.items.updateMany({sell:100},{$set:{rating:5.0}})

Delete

11. Delete item == db.intems.deleteOne({rating:3.5})

db.intems.deleteMany({})

12.db.collection.deleteOne({rating:3.5})

Shutdown or open

13. Ppppp == brew services restart mongodb-community

1. Start == mongod mac(mongosh) linux(mongo)

10 .shutdown process == exit == pgrep mongod == sudo kill 23675

8 == `db.shutdownServer()`

—--- Mongoose —--------

1. Send data to database use array {

```
// getting-started.js
const mongoose = require('mongoose');
// if not connect
main().catch(err => console.log(err));
// if connect
main().then( ()=> console.log("Data base connected
successful"))
// connect to hostname
async function main() {
 await mongoose.connect('mongodb://localhost:27017/Fas');
}
// type of data
const kittySchema = new mongoose.Schema({
   name: String
 });
// database name
const Kitten = mongoose.model('newdb-2', kittySchema);
// array send
const silence = new Kitten({ name: 'Foysal ahmed
shoun',age:20 });
silence.save()
}
```

2. Send data to database use contract From

```
// connect mongose use mongoose
main().catch((err) => console.log(err));
main().then(() => console.log("Server connect successful"));
async function main() {
 await mongoose.connect("mongodb://localhost:27017/Fas");
}
const kittySchema = new mongoose.Schema({
```

```
  name: String,
  email: String,
  addr: String,
  mess: String,
});
const Kitten = mongoose.model("fas datas", kittySchema);
// file send  new hare
app.use(express.urlencoded());
app.post("/contract", (req, res) => {
  const myData = new Kitten(req.body);
  myData.save()
Name = req.body.name;
 const cont = `Thanks ${Name} for contract us we replay to you soon
ase possible Thank's`;
 res.status(200).render("contract", { mes: cont });
});
```

—--------Github learn github—----------

*. Git config –global user,.name “Fas Foysal”

Email

*. touch .gitignore ; (node_module) (npm install)

*. Git status