بررسی نکات جاوا اسکریپت

1. یکم string method ببینیم باهم

let username = 'saeed dashti';

console.log(username.length);

console.log(username.toUpperCase());

// 12

// SAEED DASHTI

let pass = 'saeed123456'

console.log(pass.includes('123456'));

//op : true

let email = '   saeed.da@gmail.com'

console.log(email);

// op:    saeed.da@gmail.com

// clear white spaces :

console.log(email.trim());

//op:saeed.da@gmail.com

یک سری متد ها برای number

//number methods

const number = 14.679852

console.log(number.toFixed(2));

//14.68

console.log(Math.round(number));

//15

console.log(Math.floor(number));

//14  //round num to the down

console.log(Math.ceil(number));

//15  //round num to the up

تولید عدد رندوم بین 2 عدد :

const num1 = 20;

const num2 = 30;

let randomNum = Math.floor(Math.random() \* (num2 - num1) + num1);

console.log(randomNum);

1. آرایه یه مفهوم پر استفاده   
   چندین مقدار رو در یک ظرف نگهداری میکنیم

const col = ['saeed','ali',true,2,10]

console.log(col);

//['saeed','ali',true,2,10]

console.log(col.length);

//5

console.log(col[2]);

//true

const basket = ['book1','book2','book3','book4','book5','book6','book7']

const res = `you have ${basket.length} book in your basket`

const first = `the first one is : ${basket[0]}`

const sec = `the last one is : ${basket[basket.length - 1]}`

console.log(res);

console.log(first);

console.log(sec);

/\*

you have 7 book in your basket

the first one is : book1

the last one is : book7

\*/

1. Array method ها همشون مهمن یکم راجبش حرف میزنیم ولی بوقتش همشونو نیازه که بلد باشی

push(arg): arg مقداری هست که به آخر ارایه اضافه میشه!

pop(): آخرین مقدار آرایه رو کم میکنه! در واقع مقدار آخر ارایه رو ریترن میکنه

unshift(arg): arg مقداری هست که به اول ارایه اضافه میشه!

shift(): اولین مقدار آرایه رو کم میکنه! در واقع مقدار اول ارایه رو ریترن میکنه

Splice بسی مهم و کاربردیه برای حذف یا اضافه کردن عنصر در آرایه خوب بخونش تو MDN پ

forEach() :

const product = ['book1','book2','book3','book4','book5','book6'];

product.forEach((item, index)=>{

    console.log(`${index} : ${item}`);

})

/\*

0 : book1

1 : book2

2 : book3

3 : book4

4 : book5

5 : book6

\*/

\*\*\*ما معمولا تو ارایه ها دنبال سرچ کردنیم...خیلی مبحث سرچ مهمه و معمولا هم تو آرایه ها انجام میشه!

متد indexOf() : شماره index یک المنت رو در آرایه برمیگردونه ، اگر وجود نداشت منفی یک رو برمیگردونه!

arg ای که در متد قرار میگیره (indexOf(arg)) همون المنتیه که قراره تو ارایه جست و جو بشه!

const product = ['book1', 'book2', 'book3', 'book4'];

const res = product.indexOf('book3')

console.log(res);

//2

const res1 = product.indexOf('test')

console.log(res1);

//-1

خیلی مهمه این تیکه کد

با دقت بخونش میتونی بیشترم راجبش متدهاش تو MDN سرچ کنی!

const product = [{

        name: 'book1',

        id: 1,

        price: 120

    },

    {

        name: 'book2',

        id: 2,

        price: 220

    },

    {

        name: 'book3',

        id: 3,

        price: 320

    },

    {

        name: 'book4',

        id: 4,

        price: 420

    }

]

function showitem(list, prod) {

    const product = list.findIndex((item, index) => {

        return item.name === prod

    })

    console.log(product);

    //2

    return list[product]

}

const res = showitem(product, 'book3');

console.log(res);

//{ name: 'book3', id: 3, price: 320 }

روش خلاصه تر برای کد بالا :

const product = [{

        name: 'book1',

        id: 1,

        price: 120

    },

    {

        name: 'book2',

        id: 2,

        price: 220

    },

    {

        name: 'book3',

        id: 3,

        price: 320

    },

    {

        name: 'book4',

        id: 4,

        price: 420

    }

]

function showitem(list, prod) {

    return list.find((item, index) => {

        return item.name === prod

    })

}

const res = showitem(product, 'book3');

console.log(res);

//{ name: 'book3', id: 3, price: 320 }

filter(func): بسیار پرکاربرده برای آرایه ها، یک فانکشن رو به عنوان یک آرگومان میگیره

و روی تک تک مقادیر ارایه اون فانکشن رو اعمال میکنه! و ایتم هایی که بتونن اون شرط رو رد (pass) کنن وارد یک ارایه جدید میشن تو مثال بهتر هندل میشه با دقت:

const numbers = [2, 4, 12, 5, 65, 41, 4, 3];

const filterdNum = numbers.filter((num) => {

    return num >= 10

})

console.log(filterdNum);

//[ 12, 65, 41 ]

بسیار مهم :

const productItems = [{

        title: 'book1',

        exist: true

    },

    {

        title: 'book2',

        exist: false

    },

    {

        title: 'book3',

        exist: true

    },

    {

        title: 'book4',

        exist: false

    }

]

productNotExist = function (arrayItem) {

    return arrayItem.filter((item) => {

        return item.exist === false

    })

}

let res = productNotExist(productItems);

console.log(res);

//[ { title: 'book2', exist: false }, { title: 'book4', exist: false } ]

متد sort() : برای sort (مرتب سازی) کردن المنت های ارایه استفاده میشه

اگر عددی باشد به ترتیب اعداد

اگر string باشه به ترتیب حروف الفبا

تو mdn راجب مقادیر return شده مطالعه کن و ترتیب آرکومان های فانکشنه درونه سورت (a و b)

const productItems = [{

        title: 'book1',

        exist: true

    },

    {

        title: 'book2',

        exist: false

    },

    {

        title: 'book3',

        exist: true

    },

    {

        title: 'book4',

        exist: false

    }

]

const sortProducts = function (products) {

    products.sort((a, b) => {

        if (a.exist === true && b.exist === false)

            return -1;

        else if (b.exist === true && a.exist === false)

            return 1;

        else

            return 0

    })

}

sortProducts(productItems);

console.log(productItems);

/\*

[

  { title: 'book1', exist: true },

  { title: 'book3', exist: true },

  { title: 'book2', exist: false },

  { title: 'book4', exist: false }

]

\*/

این کد خیلی باحاله بخونش :

const account = {

    name: 'Saeed',

    outgo: [],

    income: [],

    addOutgo: function (desc, amount) {

        this.outgo.push({

            description: desc,

            amount: amount

        })

    },

    addIncome: function (desc, amount) {

        this.income.push({

            description: desc,

            amount: amount

        })

    },

    getAccountSummary: function () {

        let resOutgo = 0;

        let resIncome = 0;

        let balance = 0;

        this.outgo.forEach((item) => {

            resOutgo += item.amount;

        });

        this.income.forEach((item) => {

            resIncome += item.amount;

        })

        balance = resIncome - resOutgo;

        return `${this.name} ----> income : ${resIncome} and outgo: ${resOutgo} and balance : ${balance}`;

    }

}

account.addOutgo('cafe', 39);

account.addOutgo('book', 59);

account.addIncome('job', 1000);

console.log(account.getAccountSummary());

//Saeed ----> income : 1000 and outgo: 98 and balance : 902

Dom: دستورات js رو مرتبط میکنه به استک فرانت و فرایند حذف و اضافه و تغییر المنت های htmlهست

هرجا دیدم دستور js ای با document شروع شد یعنی اشاره به dom

Document یک آبجکت هستش و با . به متد هایی که داره دسترسی پیدا میکنیم

<body>

    <p>angular</p>

    <p>js book</p>

    <p>vue</p>

    <p>react js</p>

    <p>react</p>

    <script src="app.js"></script>

</body>

const ps = document.querySelectorAll('p');

ps.forEach((item) => {

    console.log(item);

    if (item.textContent.includes('js'))

        item.remove()

})

/\*

angular

vue

react

\*/

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ازین به بعد بیشتر رو کد خوانی کار میکنیم...یه توضیح کلی مینویسم و بعدش رو کد عمیق شو :

//create html element

const para = document.createElement('p')

//add content in it

para.textContent = 'is good framework for js !'

//append it to dom

document.querySelector('body').appendChild(para);

const productItems = [{

        product: 'book1',

        isExist: true

    },

    {

        product: 'book2',

        isExist: true

    },

    {

        product: 'book3',

        isExist: false

    },

    {

        product: 'book4',

        isExist: true

    },

    {

        product: 'book5',

        isExist: true

    },

    {

        product: 'book6',

        isExist: true

    },

    {

        product: 'book7',

        isExist: false

    }

]

const availableItem = productItems.filter((item) => {

    return item.isExist;

})

const h3 = document.createElement('h3');

h3.textContent = `number of availale item is : ${availableItem.length}`;

document.querySelector('body').appendChild(h3);

productItems.forEach((item) => {

    const p = document.createElement('p');

    p.textContent = item.product;

    document.querySelector('body').appendChild(p);

})

/\*

number of availale item is : 5

book1

book2

book3

book4

book5

book6

book7

\*/

دو ایونته input و change برای input های html بسیار مهم هستند  
که خوبه با سرچ بررسی شون کنی!

بخون برای فهمیدن فرم

        <form id="form">

            <input type="text" name="TitleName" autocomplete="off">

            <button>add product</button>

        </form>

document.querySelector("#form").addEventListener("submit", (e) => {

  e.preventDefault();

  console.log(e.target.elements.TitleName.value);

  e.target.elements.Title.value = "";

});

هرچی تو input نوشتیم رو لاگ میکنه  
بعد از لاگ کردن اینپوت رو خالی میکنه !

آبجکت e رو در چک باکس کامل بررسی کن

      <label for="cb">

        <input id="cb" type="checkbox" />Checkbox

      </label>

document.querySelector('#cb').addEventListener('change',(e)=>{

  console.log(e.target.checked);

})

کمی crud و مفهومش رو بررسی کن

داکیومنت زیاد داره

<https://en.wikipedia.org/wiki/Create,_read,_update_and_delete>

و همچنین localstorage

دیتاش همیشه باید string باشه موقع set شدن

زمانی که خواستیم دیتا رو get کنیم میتونیم تبدیلش کنیم