

Destructuring, Date, JSON (JavaScript Object Notation) HOME WORK

Destructuring, Date, JSON (JavaScript Object Notation)

1.

To-Do List Loyihasi Tavsifi

Obyekt yondashuvi: Biz to-do ro'yxatini boshqarish uchun obyekt yaratamiz. Bu obyekt ro'yxat elementlarini saqlash, yangi vazifa qo'shish, vazifa o'chirish va ro'yxatni konsol orqali ko'rish kabi metodlarga ega bo'lsin.

Metodlar:

- `addTask(task)`: Yangi vazifani ro'yxatga qo'shadi.
- `removeTask(index)`: Berilgan indeksdagi vazifani o'chiradi.
- `printTasks()`: Barcha vazifalarni konsolga chiqaradi

2.

Vazifa 1: Kirim va Chiqimlar Uchun Obyekt Strukturasini Yaratish

- **Maqsad:** Xo'jalik kirim va chiqimlarini saqlash uchun obyekt yaratish.
- **Ishlar:**
 - `Expenses` nomli obyekt yaratish. Bu obyektning ichida `water`, `gas`, `electricity` kabi xususiyatlar bo'ladi, ularning boshlang'ich qiymatlari 0 ga teng.
 - Har bir xarajat turiga oid metodlar qo'shiladi, bu metodlar orqali foydalanuvchi o'z xarajatlarini kirita olishi va o'chira olishi mumkin.

Misol:

javascript

``

JS

```
let Expenses = {
  water: 0,
  gas: 0,
  electricity: 0,
  addExpense: function(type, amount) {

  },
  removeExpense: function(type, amount) {

  },
  printMonthlyReport = function() {

  };
};
```

Vazifa 2: Oylik Hisobotlarni Ko'rsatish

- **Maqsad:** Har bir xarajat turining oylik umumiy summasini hisoblaydi va konsolga chiqaradi.
- **Ishlar:**
 - `Expenses` obyektiga `printMonthlyReport` metodini qo'shish.

Misol:

javascript

```
console.log(`Suv uchun oylik xarajat: ${this.water}`);
console.log(`Gaz uchun oylik xarajat: ${this.gas}`);
console.log(`Elektr energiyasi uchun oylik xarajat: ${this.electricity}`);
```

JS

3.

Harflar va Raqamlarni Hisoblash

Biror matn qabul qilib, unda nechta harf va raqam borligini hisoblaydigan funksiya yozing. Natijani obyekt sifatida qaytaring.

Misol uchun:

```
countAll("Hello World") → { "HARFLAR": 10, "RAQAMLAR": 0 }
countAll("H3ll0 Wor1d") → { "HARFLAR": 7, "RAQAMLAR": 3 }
countAll("149990") → { "HARFLAR": 0, "RAQAMLAR": 6 }
```

JS

4.

Memorial Day uchun xarid qilish!

Har bir narsa uchun qo'llaniladigan 6% soliqni hisobga olgan holda, sotib olingan mahsulotlar soniga ko'ra jami narxni hisoblovchi funksiya yarating. Misollar:

```
checkout([
  { desc: "kartoshka chiplari", prc: 2, qty: 2, taxable: false },
  { desc: "gazlangan suv", prc: 3, qty: 2, taxable: false },
  { desc: "qog'oz idishlar", prc: 5, qty: 1, taxable: true }
]) → 15.3
```

JS

Bu funksiya, sotib olingan mahsulotlarning har birining miqdori va narxi ko'rsatilgan ro'yxatni qabul qiladi va ularning umumiy qiymatini hisoblaydi. Agar mahsulot soliqa tortiladigan bo'lsa, uning narxiga 6% soliq qo'shiladi.

5. Bir funksiya yarating, bu funksiya matnlar ro'yxatini qabul qilib, faqatgina ichida raqamlar bo'lgan matnlarni saqlagan yangi ro'yxatni qaytaradi. Agar raqamlar bo'lgan matnlar bo'lmasa, bo'sh ro'yxat qaytariladi. Misol uchun:

```
numInStr(["1a", "a", "2b", "b"]) → ["1a", "2b"]
numInStr(["abc", "abc10"]) → ["abc10"]
numInStr(["abc", "ab10c", "a10bc", "bcd"]) → ["ab10c", "a10bc"]
numInStr(["this is a test", "test1"]) → ["test1"]
```

JS

BONUS

Quyida keltirilgan JavaScript dasturida CRUD ("Create, Read, Update, Delete") amallarini bajaruvchi ma'lumotlar bazasini simulyatsiya qilish uchun murakkabroq vazifa taqdim etilgan.

1. **Create (Yaratish)** funksiyasi:

- Yangi foydalanuvchi ma'lumotlarini qabul qilib, uning nomi bilan `studentsList` obyektiga qo'shing.
- Agar bu nom allaqachon mavjud bo'lsa, xatolik xabarini qaytaring.
- Aks holda, yangi foydalanuvchi qo'shilgandan so'ng, butun ro'yxatni qaytaring.

2. **Read (O'qish)** funksiyasi:

- Foydalanuvchi nomi berilganda, u nom bilan `studentsList`da qidiruv o'tkazing.
- Agar foydalanuvchi topilsa, uning ma'lumotlarini qaytaring.
- Topilmasa, "Foydalanuvchi topilmadi" kabi xabar qaytaring.

3. **Update (Yangilash)** funksiyasi:

- Foydalanuvchi nomi va yangilash kerak bo'lgan ma'lumotlar obyektni qabul qiladi.
- Agar foydalanuvchi mavjud bo'lsa, uning ma'lumotlarini yangilang va yangilangan ma'lumotlarni qaytaring.
- Foydalanuvchi topilmasa, xatolik xabarini qaytaring.

4. **Delete (O'chirish)** funksiyasi:

- Berilgan nom bo'yicha foydalanuvchini `studentsList`dan o'chiring.
- O'chirish muvaffaqiyatli bo'lsa, "Foydalanuvchi o'chirildi" kabi xabar qaytaring.
- Foydalanuvchi topilmasa, "Foydalanuvchi topilmadi" kabi xabar qaytaring.

MAPdan foydalangan holatda ishlang =