

## Home Work

## Home Work

== Kodni tushunishni osonlashtirish uchun sharh qo'ying! ==

==ASYNCHRONE VA SYNC qilib ishlang ==

1. 100ta tasodifiy sonni generatsiya qilib `sonlar.txt` fayliga yozadigan `writeNumbers` funksiyasini yozing. Bunda har bir son yangi qatorga yozilishi kerak. So'ngra, ushbu fayldan kiritilgan sonlarni o'qib olib massivga yozuvchi `readNumbers` funksiyasi yozing.
- 

2. Tasodifiy Sonlarni Faylga Yozish va O'qish

### Masala tavsifi:

Node.js dasturida ishlatiladigan ikkita funksiya yaratishingiz kerak. Birinchisi, 100 ta tasodifiy sonni `sonlar.txt` fayliga yozadigan `writeNumbers` funksiyasi. Ikkinchisi, `sonlar.txt` fayldan sonlarni o'qib olib, ularni massivga o'tkazuvchi `readNumbers` funksiyasi.

### Vazifa talablari:

---

#### 1. `writeNumbers` Funksiyasi:

- 100 ta tasodifiy son generatsiya qiladi (masalan, 1 dan 100 gacha).
  - Har bir sonni yangi qatorga `sonlar.txt` fayliga yozadi.
- 

#### 2. `readNumbers` Funksiyasi:

- `sonlar.txt` fayldan barcha sonlarni o'qiydi.
  - O'qilgan sonlarni massivga saqlaydi.
- 

#### 3. Fayl Tizimi Bilan O'yin:

- `FS` modulidan foydalanib quyidagilarni bajaradigan Node.js dasturini yozing:
    - `"my_nodejs_files"` deb nomlangan yangi katalog yarating.
    - `"my_nodejs_files"` ichida `"hello_world.txt"` deb nomlangan matnli fayl yarating, ichida `"Hello, world!"` deb yozilgan bo'lsin.
    - `"hello_world.txt"` faylining tarkibini o'qib, uni konsolga chiqaring.
    - **Bonus:** `"hello_world.txt"` fayliga hozirgi sana va vaqtni yangi qator sifatida qo'shing.
- 

#### 4. `Operation` Tizim Odiseyasi:

- `OS` modulidan foydalanib quyidagilarni bajaradigan Node.js dasturini yozing:
  - Dasturni ishga tushirgan kompyuteringizning host nomini oling.
  - Hozirgi foydalanuvchining foydalanuvchi nomini oling.

- Bu ikki ma'lumotni foydalanuvchiga do'stona formatda konsolga chiqaring (masalan, "Hostname: [hostname], Username: [username]")

---

#### 5. Yo'l Topuvchi Pro:

- `PATH` modulidan foydalanib quyidagilarni bajaradigan Node.js dasturini yozing:
  - Joriy ishchi katalogning absolyut yo'lini oling.
  - "docs" deb nomlangan ichki katalog ichida "readme.md" deb nomlangan yangi faylga nisbatan nisbiy yo'lni yarating. (Masalan, "../docs/readme.md")
  - **Bonus:** "docs" katalogi mavjudligini tekshiring. Agar mavjud bo'lmasa, "readme.md" yaratishdan oldin uni yarating.

---

#### 6. Eventful Evening :

- Bu biroz murakkabroq! `EVENTS` modulidan foydalanib sodda hodisa yuboruvchini yaratadigan Node.js dasturini yozing.
- Hodisa yuboruvchi foydalanuvchidan yangi qator kirganda "dataReceived" hodisasini chiqarishi kerak.
- Kiritilgan ma'lumotlarni konsolga yozib chiqaradigan tinglovchi funksiyani yarating.

---

#### 7. O'qish va O'rganish:

- Foydalanuvchidan `process.argv` yordamida ko'rsatilgan fayl tarkibini o'qish uchun `fs.readFile` dan foydalanadigan Node.js skriptini yozing.
- Agar fayl mavjud bo'lsa, uning tarkibini konsolda ko'rsating.
- Agar fayl mavjud bo'lmasa yoki xato bo'lsa, foydalanuvchi uchun do'stona xato xabarini berib turadigan tuzilma yarating.

---

#### 8. Yozishni Boshlaymiz! :

- "Hello World!" xabarini "message.txt" deb nomlangan yangi faylga yozadigan `fs.writeFileSync` dan foydalanadigan skript yarating.
- Potentsial xatolarni muloyimlik bilan qo'lga olish (masalan, try-catch bloki yordamida).
- Agar fayl muvaffaqiyatli yozilgan bo'lsa, konsolda muvaffaqiyat xabarini chiqaring.

---

#### 9. Qo'shish! :

- Endi bir oz ziravor qo'shamiz! " This is appended content." qo'shimcha xabarini "message.txt" fayliga qo'shadigan `fs.appendFile` dan foydalaning.
- Yana xatolarni muloyimlik bilan qo'lga olish kerak. Qo'shimcha muvaffaqiyatli bo'lganida muvaffaqiyat xabarini chiqaring.

---

#### 10. Katalogni Qazib Olish :

- Kataloglarni o'rganaylik! Joriy ishchi katalog tarkibini (fayl nomlari) o'qish uchun `fs.readdir` dan foydalaning.
- Qaytgan ro'yxat orqali aylanib, har bir fayl nomini konsolda chiqaring.

---

## BONUS

11. `create.js` - `files` papkasida `fresh.txt` nomli yangi fayl yarating, ichiga "I am fresh and young" matnini yozing. Agar fayl allaqachon mavjud bo'lsa, "FS operation failed" xabari bilan xato tashlashingiz kerak.

---

12. `copy.js` - `files` papkasini va uning barcha tarkibini `files_copy` nomli yangi papkaga nusxalang. Agar `files` papkasi mavjud bo'lmasa yoki `files_copy` allaqachon yaratilgan bo'lsa, "FS operation failed" xabari bilan xato tashlashingiz kerak.

---

13. `rename.js` - `wrongFilename.txt` nomli faylni `properFilename.md` deb qayta nomlang. Agar `wrongFilename.txt` fayli topilmasa yoki `properFilename.md` allaqachon mavjud bo'lsa, "FS operation failed" xabari bilan xato tashlashingiz kerak.

---

14. `delete.js` - `fileToRemove.txt` nomli faylni o'chiring. Agar `fileToRemove.txt` fayli topilmasa, "FS operation failed" xabari bilan xato tashlashingiz kerak.

---

15. `list.js` - `files` papkasidagi barcha fayl nomlarini konsolga chiqaring. Agar `files` papkasi mavjud bo'lmasa, "FS operation failed" xabari bilan xato tashlashingiz kerak.

---

16. `read.js` - `fileToRead.txt` faylining tarkibini konsolga chiqaring. Agar `fileToRead.txt` fayli mavjud bo'lmasa, "FS operation failed" xabari bilan xato tashlashingiz kerak.

---

Vazifalarni bajarish uchun maslahatlar:\*\*

- Node.js'dagi `fs` modulidan foydalaning. Ushbu modulni `esmodule` orqali import qilishingiz mumkin.
- Fayl va papkalarni yaratish, o'chirish, nusxalash va o'qish uchun kerakli `fs` metodlarini qo'llang.
- Xatolarni to'g'ri qo'lga olish uchun `try...catch` bloklaridan foydalaning.
- Har bir amalni alohida funksiya sifatida yozing va uni export qiling, shunda boshqa fayllardan osongina chaqirish mumkin bo'ladi.
-