NODE-JS INTRO -1

1. İlgili bağlantıdan node js kurulumunu next ,next ile kuralım . https://nodejs.org/en
2. Çalışma ortamı olarak vs-code tercih ettim herhangi bir ide kullanabilirsiniz.
3. node-js nedir ?

- Node-js java script dilini kullanarak sunucuda çalışan bir platformdur.
- chrome v8 c++ ile yazılmış java script motoru temel alınarak oluşturulmuştur .
- Genellikle web uygulamaları, API'ler, mikro hizmetler ve ağ uygulamaları gibi alanlarda kullanılır. Yüksek performansı, ölçeklenebilirliği ve geniş ekosistemi nedeniyle özellikle modern web geliştirmenin popüler bir aracıdır.
- **Java-csript nedir ?**
- **Etkileşimli** web sayfasında js motoru ile çalışan programlama dilidir.

4. Neden node-js bu kadar popüler?

- event driver olay tetiklemeli olması
- npm ile nodul çeşitliliği .NPM (Node Package Manager) tekerleği icat etmenin önüne geçerek önceden oluşturulmuş bir çok paket ile hız ve kolaylık sağlar .
- Asekron -paralel processing .Aynı anda bir çok process (işlem yapabilirliği) yüksek hız sağlıyor .Non bloking
- Çapraz platform desteği. çeişitli işletim sistemlerinde çalışabilme kolaylığı .

5.Callback Nedir?

Örneklem: meşgul bir telefona yapılan çağrıdan sonra meşguliyet durumunun ortadan kalkması durumunda vakit kaybetmeksizin iletişimin sağlanmasını sağlar .

Register callback edebilmek? sunucuya yapılan sorgu sonrasında cevap geldiğinde bilginin iletilmesi.

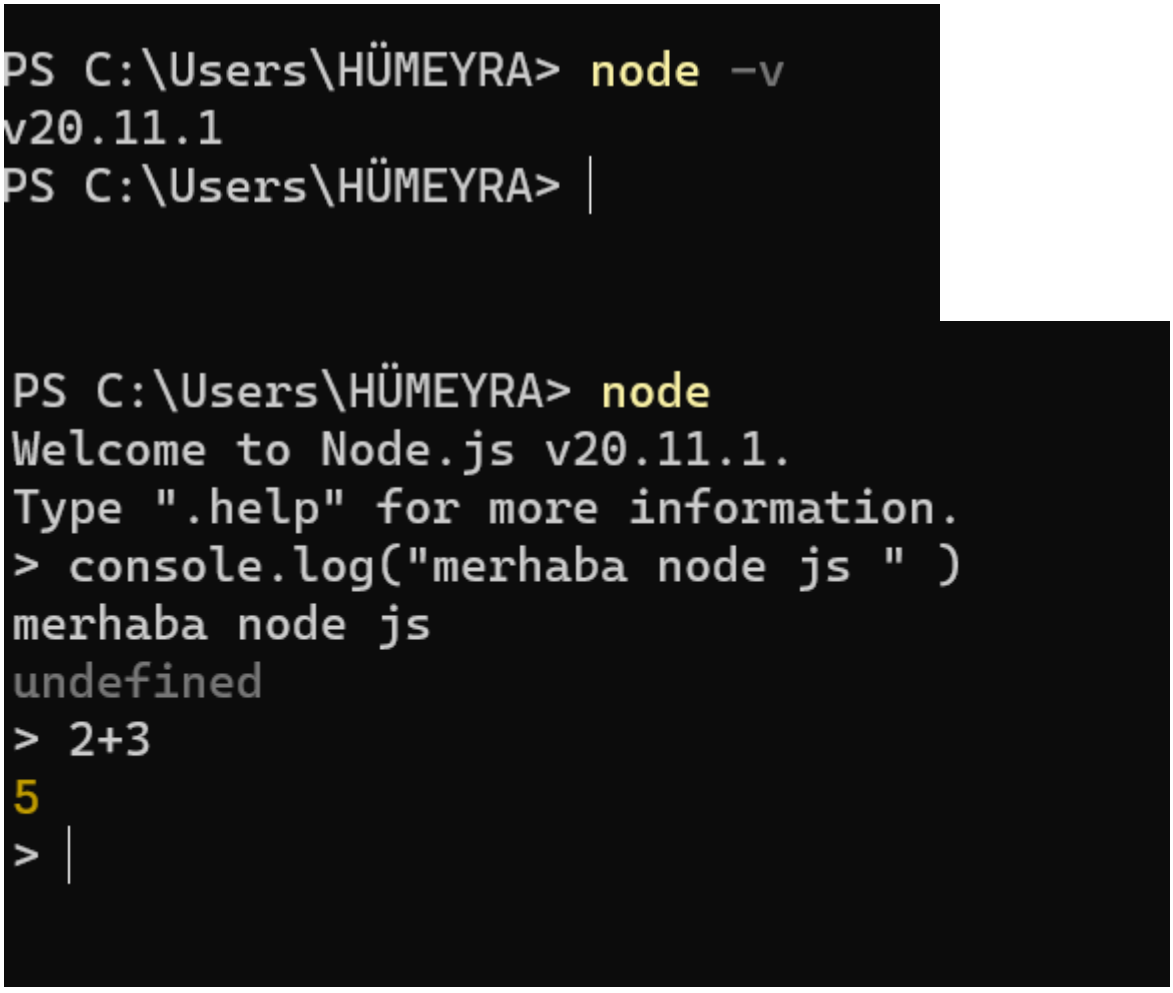
Paralel processing ile de sorgudan yanıt gelene kadar yeni bir processing işleme alınmasıdır.

```
function birinciFonksiyon(callback) {
  console.log("Birinci fonksiyon çalışıyor");
  // Simüle edilmiş bir asenkron işlem
  setTimeout(function() {
    console.log("Birinci fonksiyon tamamlandı");
    callback();
  }, 1000);
}

function ikinciFonksiyon() {
  console.log("İkinci fonksiyon çalışıyor");
}

birinciFonksiyon(ikinciFonksiyon);
```

```
Node.js v20.11.1
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\deneme> node asekron.js
Birinci fonksiyon çalışıyor
Birinci fonksiyon tamamlandı
İkinci fonksiyon çalışıyor
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\deneme>
```

6. node js yüklendi mi?

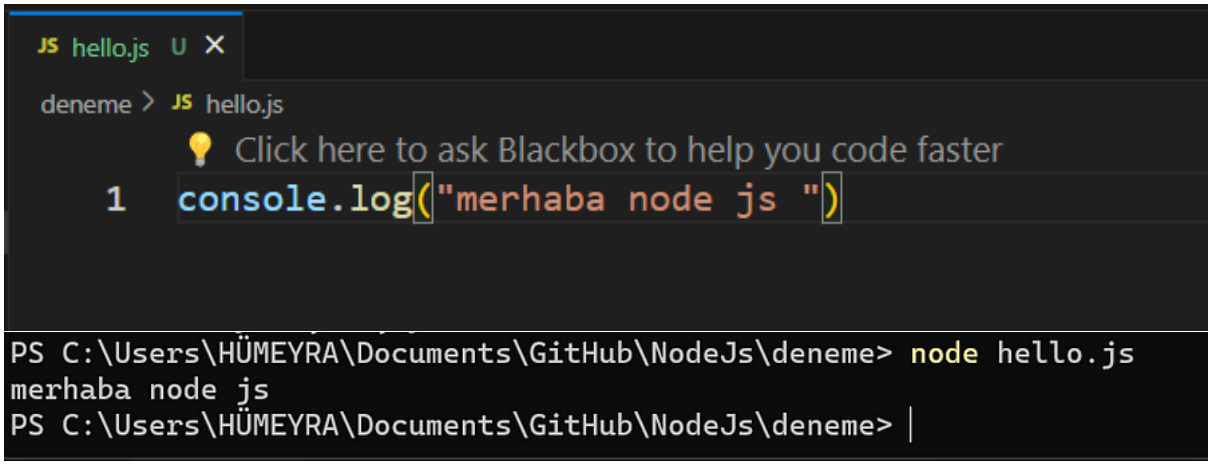
Versiyon kontrolune bakalım .



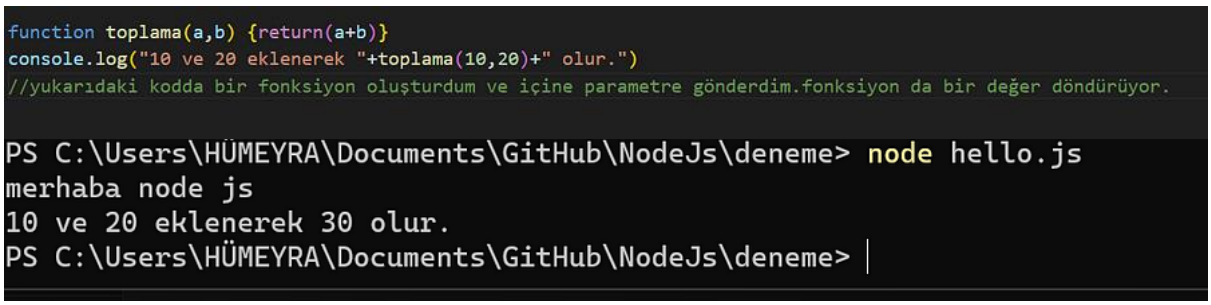
7. Bir dosya açalım “hello.js” adında ve dosya yolunu cmd ekranında “cd dosya yolu” şeklinde geçiş yapalım. İlgili js dosyasını çalıştırmak için cd ile bulunduğu dizine geliniz.

Dosya içeriğine “**console.log (‘merhaba node js ’)** ” yazdıktan sonra kaydediniz .

node “hello.js” örneğindeki gibi kendi dosya adınızı yazarak dosyayı çalıştırabilirsiniz.



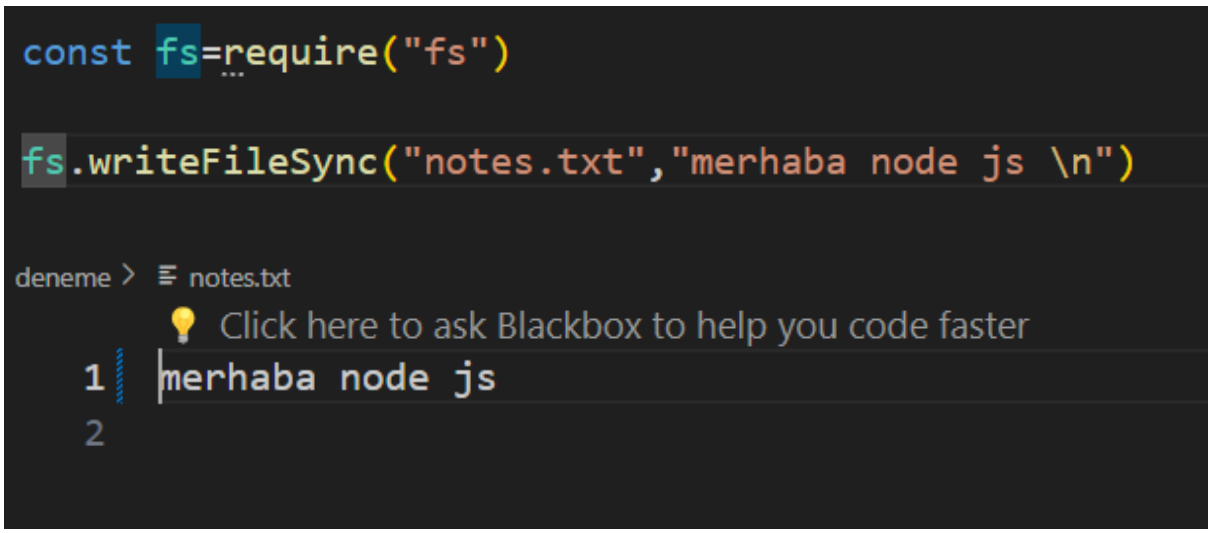
8. node -js de fonksiyon tanımlama



9.node-js require kullanımı.

- require modülleri içe aktarmak için kullanırlar .

notes.txt dosyası yoksa dosyayı açar yoksa yeni bir dosya oluşturur ve içiriğini ilgili string ile doldurur.



```
const fs=require("fs")

fs.writeFileSync("notes.txt","merhaba node js \n")

fs.appendFileSync("notes.txt","node js devam \n")
```

```
deneme > notes.txt
Click here to ask Blackbox to help you code faster
1  merhaba node js
2  node js devam
3
```

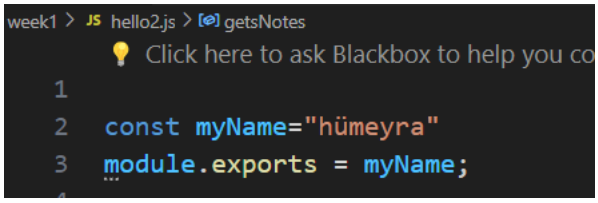
File sistem `fs` modülü ile yapılabilecek diğer işlemler şunları içerir:

- Dosya yazma: `fs.writeFile()`
- Dosya güncelleme: `fs.appendFile()`
- Dosya silme: `fs.unlink()`
- Klasör oluşturma: `fs.mkdir()`
- Klasör silme: `fs.rmdir()`
- Dosya veya klasör var mı diye kontrol etme: `fs.existsSync()`

10. Değişken çekme: hello2.js adındaki dosyada değişkeni tanımlayalım ve hello.js de bu değişkeni kullanalım.

```
deneme > JS hello2.js > ...
Click here to ask Blackbox to help you code faster
1
2  const myName="hümeyra"
3
```

```
const name=require("./he import name
console.log("myName: ", name); //bu kodla hello2.js dosyasının içindeki myName değişkenini çağırdım.
```

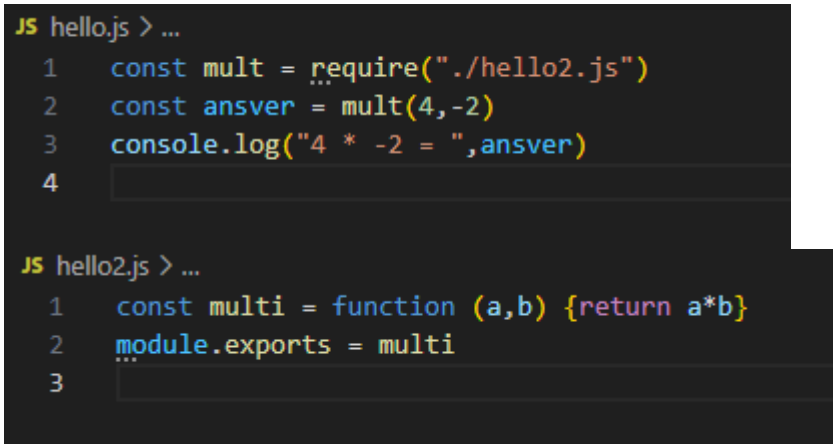


Doğrusu bu şekilde .

```
Node.js v20.11.1
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\deneme> node hello.js
merhaba node js
10 ve 20 eklenerek 30 olur.
myName:  {}
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\deneme> node hello.js
merhaba node js
10 ve 20 eklenerek 30 olur.
myName:  hümeyra
PS C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\deneme>
```

- Hata sebebi export edilmediğinde boş dönüyor.
- Export ile değişken iletiliyor.

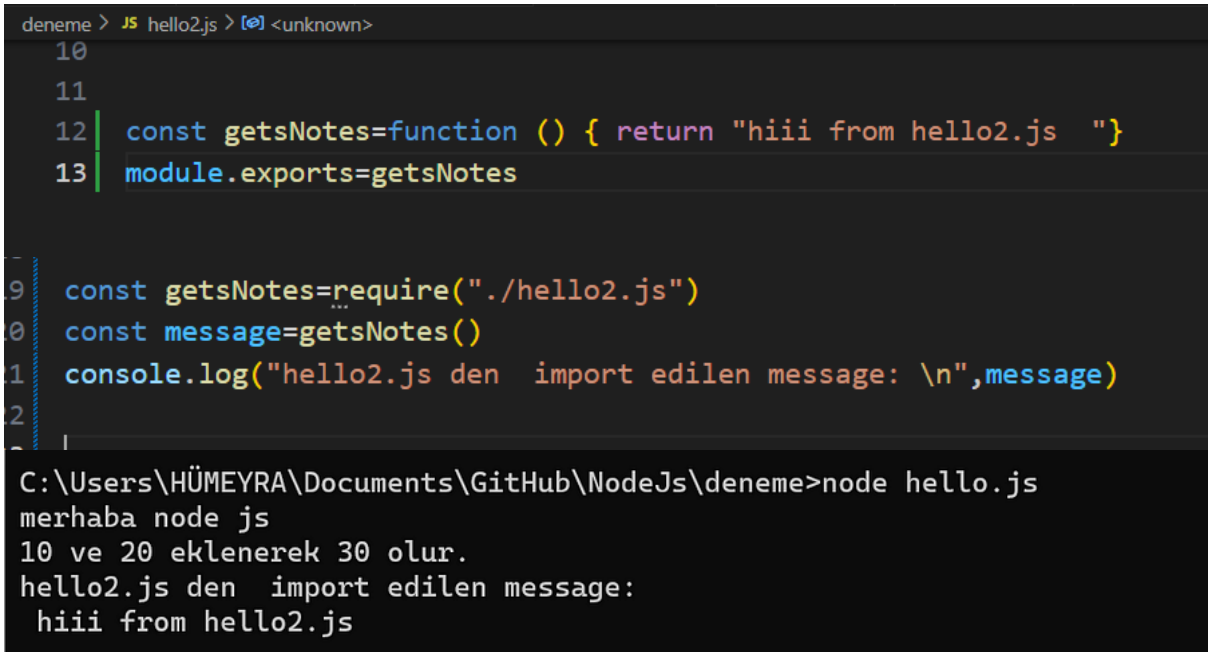
11. fonksiyon import etme.



hello2.js dosyasında fonksiyon tanımlandı ve export edildi. hello.js doyasında module içe aktarıldı ve multi fonksiyonu çağrılarak işlem sonucu consola yazıldı.

```
C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\deneme>node hello.js
merhaba node js
10 ve 20 eklenerek 30 olur.
4*-2= -8
```

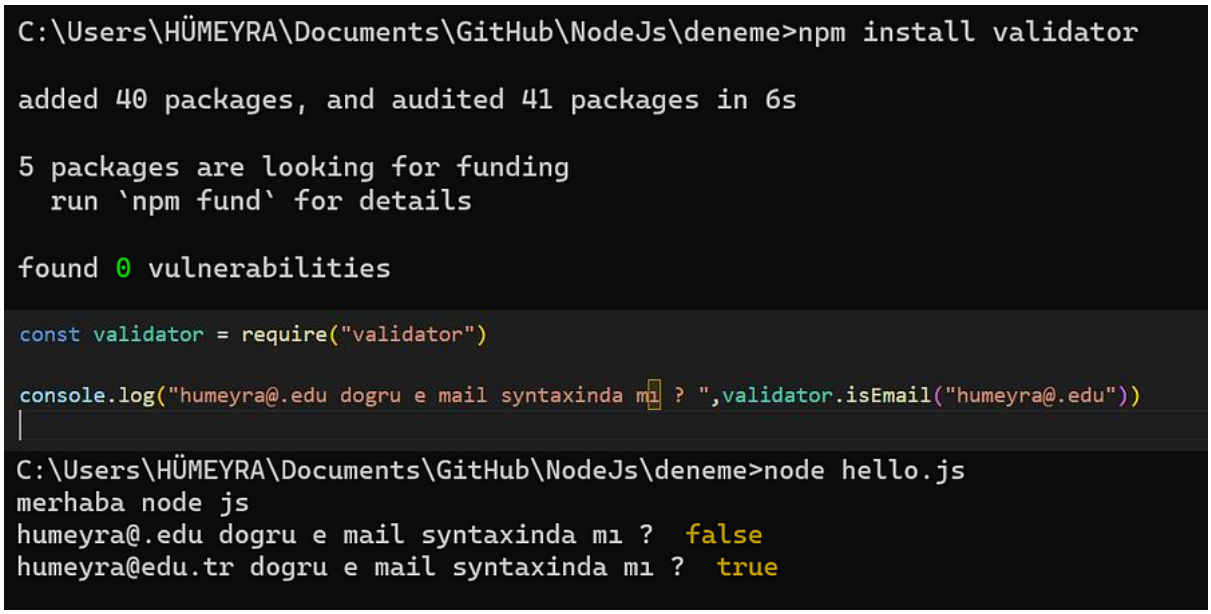
örnek: hello2.js dosya içeriğinde tanımlanan fonksiyonu hello.js içerisine import etme.



12. npm Modul **VALİDATOR** import etme.

validator nedir? `validator` kütüphanesi, e-posta adresi, URL, sayı, tarih gibi belirli türdeki verilerin doğruluğunu kontrol etmek için hazır fonksiyonlar sağlar. Ayrıca, girdileri düzenlemek veya temizlemek için de kullanışlı işlevlere sahiptir. Ayrıca `validator.escape` fonksiyonunu kullanarak HTML kodlarını temizliyoruz.

npm install validator : default olarak son sürümü ekler spesifik bir versiyon için **“ npm install validator@4.0 ”** gibi versiyon belirterek kullanılır



```
console.log(validator.isURL('www.btu.edu'))
```

```
C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\deneme>node hello.js
merhaba node js
BU Bir URL midir?  true
```

13. chalk modu import etme: chalk, Node.js ortamında renkli yazıları konsol üzerinde görüntülemek için kullanılan bir kütüphanedir.

```
C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\deneme>npm install chalk@4.0

up to date, audited 41 packages in 921ms

5 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

```
const chalk=require("chalk");
console.log(chalk.greenBright("bu yazı yeşil renkde gözükecektir."))
```

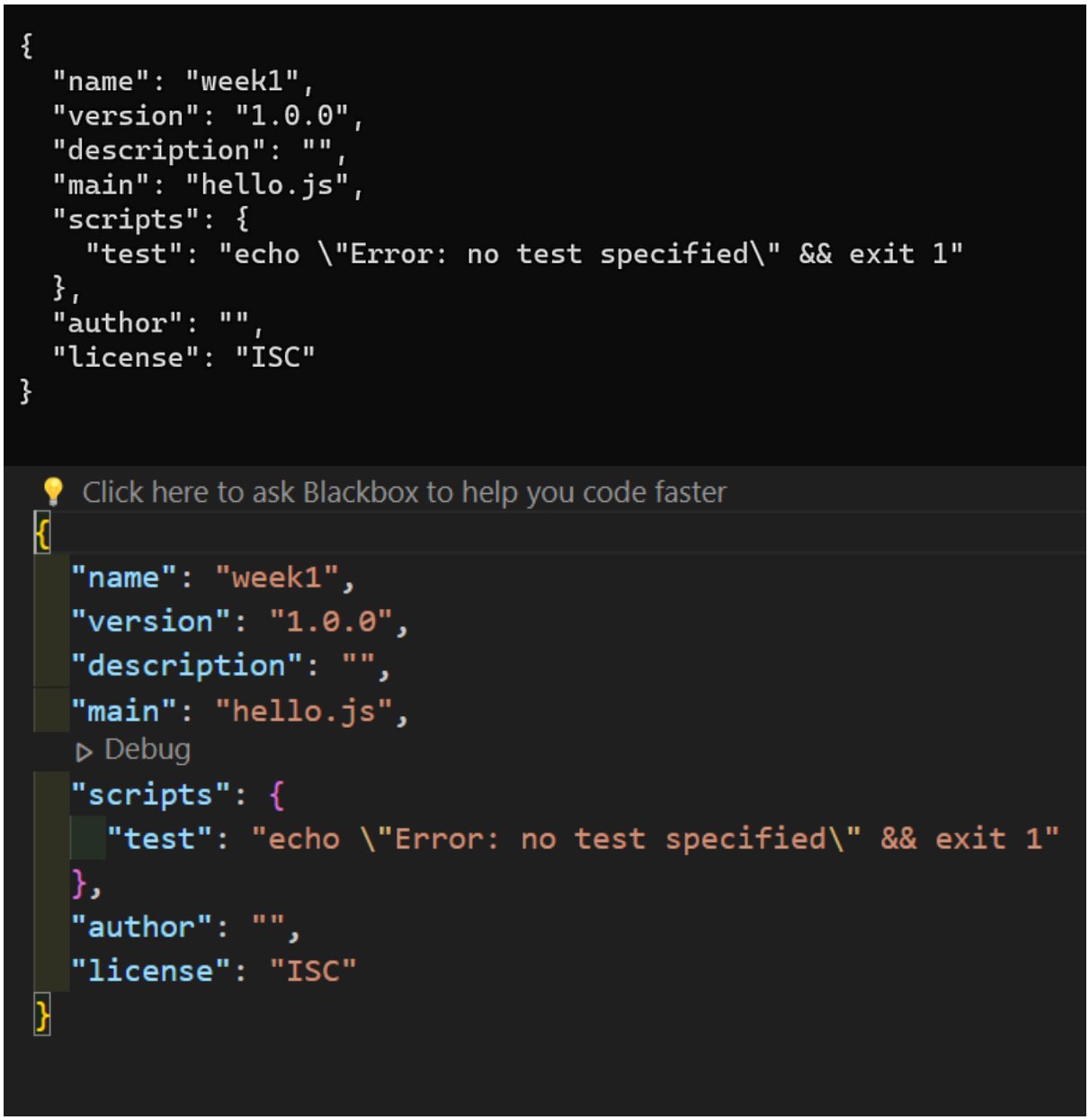
```
C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\deneme>node hello.js
merhaba node js
bu yazı yeşil renkde gözükecektir.
```

14. Komut `npm install nodemon -g` kullanılarak Node.js projelerinde `nodemon` aracını küresel olarak yüklersiniz.-g global olmasını sağlar

```
C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\deneme>npm install nodemon -g

added 33 packages in 6s

3 packages are looking for funding
  run `npm fund` for details

C:\Users\HÜMEYRA\Documents\GitHub\NodeJs\week1>nodemon hello.js
[nodemon] 3.0.3
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node hello.js`
merhaba node js
bu yazı kırmızı renkde gözükecektir.
[nodemon] clean exit - waiting for changes before restart
[nodemon] restarting due to changes...
[nodemon] starting `node hello.js`
merhaba node js
bu yazı mavi renkde gözükecektir.
[nodemon] clean exit - waiting for changes before restart
```

nodemon ile çalışılan dosya üzerinde tekrar tekrar node hello.js çalıştırılmasına gerek kalmadan yapılan her değişiklik otomatik olarak yansıtılır .

15. npm init ile yazılan js dosyası package.json dosyasına dönüştürülür. **`npm init`**



Yukarıda default adlandırmalar ile oluşturulmuş bir package-json dosyası içeriği.

package.json nedir ?

`package.json`, bir Node.js projesinin kök dizininde bulunan ve projenin bağımlılıklarını, betiklerini (scripts), proje bilgilerini ve diğer konfigürasyonları tanımlayan bir konfigürasyon dosyasıdır. Bu dosya, bir Node.js uygulamasının yapısını ve gerekli ayarları belirler.

Genellikle `npm init` komutu kullanılarak oluşturulan `package.json` dosyası, projenin ismini, sürümünü, ana dosyasını, bağımlılıklarını, test betiklerini ve daha birçok bilgiyi içerir. Bu dosya, projenin başkaları tarafından anlaşılmasını ve kullanılmasını kolaylaştırır.

Uygulama kodları için https://github.com/hmyrcmn/NodeJs aderisine gidiniz.

HÜMEYRA ÇİMEN