

Hafta 1

Rapor

□ Kurulum

Yapmamız gereken ilk iş Node.js'in yüklü olup olmadığını anlamak. Bunun için başlat sekmesine tıklayıp "cmd" yazmamız ve terminali açmamız gerekmektedir. Açılan terminale "node -v" yazınız. Ekranda "v20.11.1" formatında bir yazı çıkmadı ise aşağıdaki indirme adımlarını takip ediniz. Hali hazırda bilgisayarında Node.js olanlar bir paragraf atlayarak okumaya devam edebilirler.

Tarayıcınıza girin. Node.js indir yazınız ve "nodejs.org/en/download" siteye giriniz. Kullandığınız işletim sistemine bağlı olarak en son sürümünü indiriniz. Uygulamayı kurunuz.

□ Peki nedir bu Node.js?

JavaScript, diğer çoğu programlama dili gibi sadece tarayıcıda çalışan, tarayıcının izin verdiği ölçüde ve makinenizden bağımsız bir işlem olarak çalışır.

Node.js bir programlama dili değildir. Node.js, sunucu tarafında JavaScript kodunu çalıştırmanıza olanak tanır. Bu sayede dosya sistemine erişebilir ve veritabanlarına bağlanabilirsiniz. Örneğin, kullanıcının tarayıcısında değil, sunucuda çalışan JavaScript kodu aracılığıyla butona tıklama olayını işleyip, yönlendirme işlemlerini gerçekleştirebilirsiniz.

Node.js, Chrome'un V8 JavaScript motoru üzerine inşa edilmiş bir JavaScript çalıştırma ortamıdır. V8, Google'ın açık kaynaklı bir projesidir. Bu motor, JavaScript kodunu alır ve bilgisayarınızın yürütebileceği makine koduna derler. Hem Chrome hem de Node.js C++ ile yazılmıştır. Fonksiyonlar ve nesnelere özel işlevlik kazandırır.

Chrome ve Node.js, JavaScript kodunu V8'e ileterek sonuçları geri alırlar.

Chrome'a giriniz. Sayfanın herhangi bir yerine tıklayınız ve "incele" seçeneğine tıklayınız. Açılan pencere de console kısmına geliniz. Terminali açınız. Terminale ve console'a 2+3 yazınız. Her ikisinde de 5 sonucunu elde etmelisiniz. Peki bir örnek daha deneyelim. "furkan".toUpperCase() yazınız. 'FURKAN' sonucunu elde etmelisiniz.

Her ikisi de aynı sonuçları veriyor ama neden Node.js daha çok tercih ediliyor?

Node.js, olay odaklı, bir uygulama yanıt beklerken diğer başka uygulamalarda başka işlem yapılabilmesi sebebiyle hafif ve verimli olması kaçınılmaz olur. Bu da daha çok

tercih edilmesinin en temel sebebidir. Uygulamanın yanıt vermesini beklerse zaman açısından verimlilik çok kötü olurdu. Olay odaklı olması callbackleri kaydetmek ve bunları IO tamamlandığında çağırmak (call etmek)demektir.

npm, en büyük açık kaynak ekosistemidir ve Node.js ile birlikte bilgisayarınıza kurulur. Araştırmak isteyenler için mevcut paketleri internet sitesinde bulabilirsiniz.

Masaüstüne dönüp node adlı bir klasör oluşturunuz. Klasörün içine ise index.js adlı bir belge oluşturunuz.ve o belgenin içine giriniz.

KOD

Çıktı

```
console.log('Hello Node.js!')
```

Hello Node.js!

```
fs.writeFileSync('notes.txt', 'I live in Bursa')
```

Bu kodu yazarsanız hata alırsınız. çünkü ilk önce "fs" tanımlamanız gerekmektedir.

Tanımlamak için yandaki kodu yazmalısınız. --> `const fs = require('fs')`

`fs.writeFileSync('notes.txt', 'I live in Bursa')` kodu tekrardan çalıştırırsanız bu sefer bir `notes.txt` dosyayı oluşturulur ve içine 'I live in Bursa' yazar.

Bu dosyanın üstüne bir şeyler eklemek istiyorsanız,

```
fs.appendFileSync('notes.txt', ' I live in Bursa')
```

`notes` dosyasında 2 defa I live in Bursa yazılı olur

`utils.js` ve `app.j` oluşturunuz. dosyanın içine `console.log('Hello utils.js!')` yazınız. Çalıştırdığınızda herhangi bir çıktı alamayacaksınız çünkü `'utils.js'` dosyası henüz başka bir yerde çağrılmamıştır.

'`app.js` dosyasının en üstüne `require('./utils.js')` ifadesini ekleyin. Bu, `utils.js` dosyasını içeri aktaracaktır. kaydedin ve tekrar çalıştırın. artık çalışacaktır çünkü şimdi `require('./utils.js')` ifadesi en üstte bulunmaktadır.

```
const name='Cem'
```

```
console.log(name)
```

bu kod satırlarını utils.js'in içine yazın ve çalıştırın. Hata alacaksınız çünkü name değişkenine app.js ulaşamıyor. module.exports = name kodunu utils.js dosyasına yazmanız gerekmektedir. Yazıp çalıştırdığınız zaman name değişkenine erişilir ve name yazdırılır.

□ Npm nedir?

Npm; Node Package Manager ya da Node Packaged Modules olarak da denmektedir. Isaac Z. Schlueter tarafından tamamen JavaScript dili kullanılarak geliştirilmiştir. Npm temel olarak 3. parti yazılımları yüklemeyi sağlayan bir araçtır. npm, Node.js ekosisteminde yaygın olarak kullanılan açık kaynaklı bir araçtır ve kullanıcılara kullanabilecekleri paketlere erişim sağlar. Kullanıcıların hali hazırda var olan paketleri kullanmasını sağlar bu sayede işi hızlandırır.

Terminale girip “npm -v” yazar iseniz karşınızda hali hazırda bi npm olduğunu görebilirsiniz. Node.js ile beraberinde indirildiği bilgisi verilmişti.

“npm init” yazıp npm oluşturabilirsiniz. Bu bir package.json dosyası oluşturur. Genel olarak oluşturulan npm'in bağımlılıklarını gösteren bir dosyadır.

Npm, paketlerin düzenlenmesini sağlayan komutlarda barındırır. Örneğin, ‘npm install <paket-adı>’ ile bir paket yüklenebilir, ‘npm update <paket-adı>’ ile bir paket güncellenebilir ve ‘npm uninstall <paket-adı>’ ile bir paket kaldırılabilir.

Kod	Görev
npm install <paket-adı>	Paket yükler
npm uninstall <paket-adı>	Var olan paketi siler
npm update <paket-adı>	Paketi Günceller

- **Validatör**, veri doğrulama işlemlerinde kullanılır. Ancak bunu install etmemiz gerekir. İndirmek için ise “npm install validator@” kodunu terminale yazılmalıdır. Genellikle e-posta, URL gibi bazı verilerin doğrulanmasını sağlar.

Bir örnek ile validatörü pekiştirelim. app.js dosyasına gidiniz.

```
const validator= require('validator')
```

```
console.log(validator.isEmail(furkan@btu.edu')) kod
```

satırlarını yazınız.

Terminalde app.js'in bulunduğu dosyaya giriniz. Ve "node app.js" komutu ile app.js'i çalıştırınız. 'true' sonucunu alırsınız çünkü girilen mail türü doğrudur. Örnek olarak console.log(validator.isEmail(furkanbtu.edu)) böyle bir kod satırı yazılsa idi 'false' çıkışı alırdık.

- **Peki metinleri renklendirmek istersek?**

Renkli metinler için kullanılan modülün adı Chalk modülüdür. Nasıl kurulur ona bir göz atalım. Terminale "npm install chalk" yazılır ve chalk modülü kurulur. Sırada nasıl kullanıldığını öğrenmek var. Yeni bir dosya oluşturunuz. İsmine chalk.js veriniz. const chalk = require("chalk") .

```
console.log(chalk.green.bold("Success!"))
```

Yazınız ve çalıştırınız. 'Success' çıktı böyle olmalıdır. Eğer ki hata alıyorsanız

"npm install chalk@4" kodunu yazıp const chalk = require("chalk") .

console.log(chalk.green.bold("Success!")) bu kod satırlarını tekrardan çalıştırınız.

Sonuç başarılı olacaktır.

Validatör ve chalk yerel modüllerdi. Bir de global modüller var. Neden ayırım var diye merak edeceksinizdir. Local npm'ler proje yüklenir ve projeye hastır. Projenin bağımlılıklarını ölçer ve barındırır. Global npm'ler ise genele yüklenir ve bütün projeler tarafından erişilebilir. Global modüller projelerin dosyalarını değiştirmez bağımlılıklarını etkilemezler.

Global bir modül olan **nodemon** modülünü indirelim.

"npm install nodemon -g" yazarak indirilir. Eğer ki hata verir ise kodun başına "sudo" ekleyerek yani "sudo npm install nodemon -g" yazıp çalıştırırsak veya yönetici olarak çalıştırırsak sorun ortadan kalkacaktır. İndip inmediğini kontrol etmek istiyorsanız ise

"nodemon -v" komutu ile kontrol edebilirsiniz.

"nodemon app.js" ile modülü app.js dosyasına dahil edebilirsiniz. App.js yerine istediğiniz dosyaya dahil edebilirsiniz