

Furkan Kazım Çam

21360859036

Merhabalar, Bugün terminal üzerinden kullanıcıdan girdi almayı öğrenerek eğitime devam ediyoruz.

“bolum2” adında klasör oluşturup içine app.js dosyasını oluşturarak işleme başlayalım.

app.js dosyamızın içine `console.log("mehmet")` yazıp. `node app.js` kodu ile çalıştırırsanız mehmet yazacaktır. Kullanıcıdan girdi almak istediğimizde ne yapacağız bir göz atalım.

`node app.js` Furkan kodunu çalıştırırsanız yine Mehmet çıktısı alırsınız. Nedeni ise evet bir veri gönderimi var fakat app.js sayfasında veriyi karşılayan herhangi bir kod satırı yok.

`console.log(process.argv)` komutunu app.js dosyasına yazınız. Ve çalıştırın

```
PS C:\Users\furka\OneDrive\Masaüstü\bolum2> node app.js Furkan
[
  'C:\\Program Files\\nodejs\\node.exe',
  'C:\\Users\\furka\\OneDrive\\Masaüstü\\bolum2\\app.js',
  'Furkan'
]
```

Bu çıktıyı almanız gerekmektedir. Eğer çıktıyı alıyorsanız terminal üzerinden kullanıcı tarafından girdiyi alabildiğiniz demektir. Yeşil ile yazılanlar ne anlama geliyor onlara bir bakalım.

'C:\\Program Files\\nodejs\\node.exe' → node.js'in bilgisayarınızdaki konumunu gösteren satır.

'C:\\Users\\furka\\OneDrive\\Masaüstü\\bolum2\\app.js' → O an üzerinde çalıştığınız dosyanın bilgisayarınızdaki konumudur.

'Furkan' → Alınan girdilerin oluşturduğu girdi vektörüdür.

Sadece girdiyi yazdırmak için ise `console.log(process.argv[2])` komutu ile çalıştırabilirsiniz.

```
PS C:\Users\furka\OneDrive\Masaüstü\bolum2> node app.js Furkan
Furkan
```

Bugüne kadar öğrendiklerimiz ile bir not uygulaması yapmaya çalışalım.

```
const command =process.argv[2];

//Alınan girdi command değişkenine eşitlenir

if(command=="add"){

    console.log("adding note")

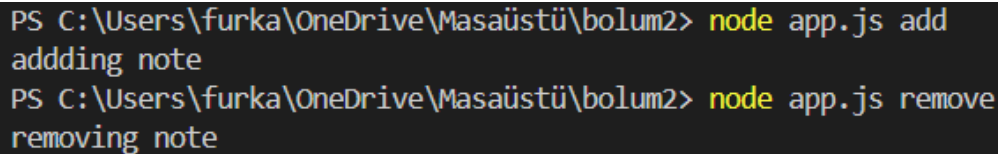
}

else if(command=="remove"){

    console.log("removing note");

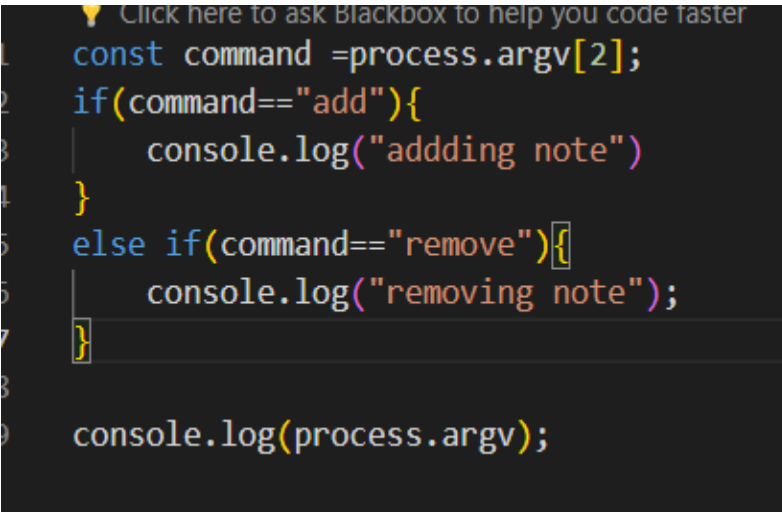
}
```

Kod satırlarını app.js'in içine yazıp alttaki gibi çalıştırırsanız aynı çıktıyı almalısınız. Burada alınan girdi command değişkenine atanır ve if else blokları ile girdi kontrol edilir.

A terminal window with a dark background. The prompt is 'PS C:\Users\furka\OneDrive\Masaüstü\bolum2>'. The first command is 'node app.js add', followed by the output 'adding note'. The second command is 'node app.js remove', followed by the output 'removing note'.

```
PS C:\Users\furka\OneDrive\Masaüstü\bolum2> node app.js add
adding note
PS C:\Users\furka\OneDrive\Masaüstü\bolum2> node app.js remove
removing note
```

Şu an yaptığımız sadece ekrana yazı yazdırmaktan ibaret. Şimdi ise oluşturulan notlara başlık ve diğer bazı özellikleri nasıl veririz öğrenelim. Bu gibi özellikleri de terminal üzerinden girdi olarak alacağımız için console.log(process.argv) komutunu app.js dosyasına eklemelisiniz. Son durumda app.js dosyamızda

A code editor window with a dark background. It shows the same code as before, but with an additional line at the bottom: 'console.log(process.argv);'. Line numbers 1 through 9 are visible on the left side of the code.

```
1 const command =process.argv[2];
2 if(command=="add"){
3     console.log("adding note")
4 }
5 else if(command=="remove"){
6     console.log("removing note");
7 }
8
9 console.log(process.argv);
```

Bu kodlar olmalıdır.

Nota başlık eklemek için

```
]
PS C:\Users\furka\OneDrive\Masaüstü\bolum2> node app.js add --title="Başlık"
adding note
[
  'C:\\Program Files\\nodejs\\node.exe',
  'C:\\Users\\furka\\OneDrive\\Masaüstü\\bolum2\\app.js',
  'add',
  '--title=Başlık'
]
```

Bu kod ile başlık ekleyeceğiz fakat şu an eklediğimiz başlığa erişemiyoruz. Erişmek için diziden çıkartmamız gerekir. Bunu hali hazırda bize sunan bir paket olduğu için bu paketi indirelim. Paketimizin adı yargs'dır.

Geçen hafta öğrendiğimiz npm modülleri nasıl indirilir dersinden de bildiğimiz üzere paketimizi indirelim.

Terminale "npm init" yazarak npm'i başlatalım. Birkaç saniye beklemenin ardından bize bazı bilgiler soracaktır. Npm hakkındaki bazı bilgilerdir. İsteğe bağlı doldurulabilir.

Bütün sorulara enter basarak geçmemizin zararı yoktur. Soruları aştıktan sonra "npm i yargs" kodu ile yargs paketimizi indirelim. Bu paket sayesinde verilere yani girdilere erişim sağlandı. Erişilen verileri kullanalım.

Yargs kütüphanesini projemize dahil edelim.

const yargs=require("yargs") app.js dosyamızın içine yazıyoruz.

```
const yargs=require("yargs")
const command =process.argv[2];
if(command=="add"){
  console.log("adding note")
}
else if(command=="remove"){
  console.log("removing note");
}

console.log(yargs.argv);
```

```
PS C:\Users\furka\OneDrive\Masaüstü\bolum2> node app.js --title="Furkan Kazım Çam"
{ _: [], tittle: 'Furkan Kazım Çam', '$0': 'app.js' }
```

Artık girdilere erişiyoruz ve başlık verebiliyoruz.

Yargs'ın bize sunduğu seçenekleri gösterir.

```
PS C:\Users\furka\OneDrive\Masaüstü\bolum2> node app.js --help
Seçenekler:
  --help      Yardım detaylarını göster [boolean]
  --version   Versiyon detaylarını göster [boolean]
```

node app.js --version komutunu deneyelim. Bu bize yargs versiyonunu verir.

```
PS C:\Users\furka\OneDrive\Masaüstü\bolum2> node app.js --version
1.0.0
```

Kodumuza yargs'ın versiyonunu ekleyelim.

Kodumuzda ki if else kısımlarını silelim. Neden diye soracak olursanız yargs bizim için bu işlemi kendi parametreleri için yapacaktır.

```
1  const yargs=require("yargs")
2  yargs.version("1.1.0")
3
4  yargs.command({
5    command:"add",
6    describe:"Yeni not eklenecek",
7    handler:function(){
8      console.log("Yeni not ekleniyor...")
9    }
10 });
11 yargs.parse();
```

Buradaki kod önceki koddaki if satırına denk gelmektedir.

“command” → alınan girdi

“describe” → kod bloğunu kısaca açıklar

“handler” → komuta özel fonksiyonlar gibi seçeneklerin bulunduğu kısım

```
PS C:\Users\furka\OneDrive\Masaüstü\bolum2> node app.js add
Yeni not ekleniyor...
```

Bu kod ile app.js'in çalıştığını görebilirsiniz.

Şimdi not ekleyebiliyoruz. Peki silme listeleme ve not okuma işlemlerinin kodunu yazalım.

Click here to ask Blackbox to help you code faster

```
const yargs=require("yargs")
yargs.version("1.1.0")

yargs.command({
  command:"add",
  describe:"Yeni not eklenecek",
  handler:function(argv){
    console.log("Yeni not ekleniyor...")
  }
})
yargs.command({
  command:"remove",
  describe:"Not silinecek",
  handler:function(argv){
    console.log(`${argv}Not siliniyor...")
  }
})
yargs.command({
  command:"list",
  describe:"Notlar listelenecek",
  handler:function(argv){
    console.log("Notlar listeleniyor...")
  }
})
yargs.command({
  command:"read",
  describe:"Not okunacak",
  handler:function(argv){
    console.log("Not okunuyor...")
  }
})
yargs.parse();
```

Komutlarımızın eklenip eklenmediğini kontrol etmek için “node app.js –help” yazabilirsiniz.

```

Yeni not ekleniyor...
PS C:\Users\furka\OneDrive\Masaüstü\bolum2> node app.js --help
app.js [komut]

Komutlar:
  app.js add      Yeni not eklenecek
  app.js remove   Not silinecek
  app.js list     Notlar listelenecek
  app.js read     Not okunacak

Seçenekler:
  --help      Yardım detaylarını göster [boolean]
  --version   Versiyon detaylarını göster [boolean]

```

```

at node:internal/main/run_main_module:28:49

Node.js v20.11.1
PS C:\Users\furka\OneDrive\Masaüstü\bolum2> node app.js add
{ _: [ 'add' ], '$0': 'app.js' }
Yeni not ekleniyor...
PS C:\Users\furka\OneDrive\Masaüstü\bolum2> node app.js remove
{ _: [ 'remove' ], '$0': 'app.js' }
Not siliniyor...
PS C:\Users\furka\OneDrive\Masaüstü\bolum2> node app.js list
{ _: [ 'list' ], '$0': 'app.js' }
Notlar listeleniyor...
PS C:\Users\furka\OneDrive\Masaüstü\bolum2> node app.js read
{ _: [ 'read' ], '$0': 'app.js' }
Not okunuyor...

```

Kodumu çalışıyor...

Command'ın diğer özelliği olan builderı kullanalım.

```

yargs.command({
  command: "add",
  describe: "Yeni not eklenecek",
  builder: {
    title: {
      describe: "Not başlığı"
    },
  },
  handler: function(argv) {
    console.log("Yeni not ekleniyor...", argv)
  }
})

```

add için builder eklendi. Bunu çalıştırdığınızda hata vermez çünkü gönderilen girdi zorunlu olup olmadığını belirtmıyoruz. bu işlevi sağlamak için builder'ın title özelliğine "demandOption: true" kodunu girmelisiniz.

```

yargs.command({
  command:"add",
  describe:"Yeni not eklenecek",
  builder:{
    title:{
      describe:"Not başlığı",
      demandOption: true
    },
  },
},

  handler:function(argv){
    console.log("Yeni not ekleniyor...",argv)
  }
})

```

Hatalı ve doğru yazımlar alttaki fotoğrafta mevcuttur.

```

PS C:\Users\furka\OneDrive\Masaüstü\bolum2> node app.js add
app.js add

Yeni not eklenecek

Seçenekler:
  --help      Yardım detaylarını göster           [boolean]
  --version   Versiyon detaylarını göster          [boolean]
  --title     Not başlığı                          [zorunlu]

Eksik zorunlu argüman: title
PS C:\Users\furka\OneDrive\Masaüstü\bolum2> node app.js add --title="furkan"
Yeni not ekleniyor... { _: [ 'add' ], title: 'furkan', '$0': 'app.js' }

```

Şimdi ise title özelliğine veri tipi verelim.

```

yargs.command({
  command:"add",
  describe:"Yeni not eklenecek",
  builder:{
    title:{
      describe:"Not başlığı",
      demandOption: true,
      type:'string'
    },
  },
},

  handler:function(argv){
    console.log("Yeni not ekleniyor...",argv)
  }
})

```

Bu kodları diğer commandlar içinde yazınız.

Command'ın diğer özelliği olan body'i kullanalım.

Body'i de builderdaki gibi aynı şekilde yazıyoruz. Aynı özellikler burda da geçerli.

```

3
4  yargs.command({
5      command:"add",
6      describe:"Yeni not eklenecek",
7      builder:{
8          title:{
9              describe:"Not ekleme başlığı",
10             demandOption: true,
11             type:'string'
12         },
13     },
14 },
15 body:{
16     describe:"Add body",
17     demandOption:true,
18     type:"string"
19 },
20
21 handler:function(argv){
22     console.log("Yeni not ekleniyor...",argv.title)
23     console.log("body",argv.body)
24 }
25 })

```

Bunu diğer commandlar içinde yapınız.

```

PS C:\Users\furka\OneDrive\Masaüstü\bolum2> node app.js add --title="furkan" --body="kazım"
Yeni not ekleniyor... furkan
body kazım
PS C:\Users\furka\OneDrive\Masaüstü\bolum2> node app.js read --title="furkan" --body="kazım"
Not okunuyor... furkan
body kazım
PS C:\Users\furka\OneDrive\Masaüstü\bolum2> node app.js remove --title="furkan" --body="kazım"
$Not siliniyor... furkan
body kazım
PS C:\Users\furka\OneDrive\Masaüstü\bolum2> node app.js list --title="furkan" --body="kazım"
Notlar listeleniyor... furkan
body kazım

```

Bu çıktıları almalısınız.

Eğer ki title veya body'i tek başına girdi olarak vererseniz hata alıacaksınız çünkü iki özelliğinde demandOption özelliğini true yaptık.

```

body undefined
PS C:\Users\furka\OneDrive\Masaüstü\bolum2> node app.js list --title="furkan"
Notlar listeleniyor... furkan
body undefined

```

Girdiler Json dosya biçiminde nasıl saklanır?

Yeni bir dosya açınız ve ismine notes-app.js veriniz. Book adında bir değişken oluşturup içine title ve author değişkenlerini girelim. Ve bunu console'a yazdıralım


```

1  const book = {
2    title: 'my title',
3    author: 'can bonomo'
4  }
5  const bookJSON = JSON.stringify(book)
6  console.log(bookJSON.title);

```

```

PS C:\Users\furka\OneDrive\Masaüstü\bolum2> node notes-app.js
{"title":"my title","author":"can bonomo"}

```

Görüldüğü üzere book objesinin özelliklerini yazdırdık şimdi ise bunlar JSON formunda nasıl yazarız ona bir göz atalım.

```

const fs = require('fs')
const book = {
  title: 'my title',
  author: 'can bonomo'
}

const bookJSON = JSON.stringify(book)
fs.writeFileSync('notes.app.json', bookJSON)
const parsedObject = JSON.parse(bookJSON)
console.log(parsedObject.title);

```

Bir dosyaya yazma işlemi olduğu için bir fs tanımlanır

Book'un özellikleri json formatına dönüştürülür

Notes-app.json dosyasına yazılır.

Console'da book'un başlığı yazdırılır

```

PS C:\Users\furka\OneDrive\Masaüstü\bolum2> node notes-app.js
my title

```

Click here to ask Blackbox to help you code faster

```
1 const fs=require('fs')
2 const book={
3   title:'my title',
4   author:'can bonomo'
5 }
6
7
8 const bookJSON=JSON.stringify(book) // {"title":"Kitap başlığım","author":"Kitap yazar"}
9 fs.writeFileSync('notes-app.json',bookJSON)
10 const parsedObject=JSON.parse(bookJSON) // { title: 'Kitap başlığım', author: 'Kitap yazar' }
11 console.log(parsedObject.title, parsedObject.author);
12 const databuffer=fs.readFileSync('notes-app.json')
13 const dataJSON=console.log(databuffer.toString())
14 const data=JSON.stringify(dataJSON)
15 console.log(data);
```

```
PS C:\Users\furka\OneDrive\Masaüstü\bolum2> node notes-app.js furkan
my title can bonomo
{"title":"my title","author":"can bonomo"}
```