

Gizem Avcı 21360859071

NodeJS ile Web Programlama Hafta-2 Raporu

Bu haftaki derste komut satırından argüman alma görüldü. Bunu npm modülü kullanılmadan ve npm modülü kullanılarak uygulandı.

Öncelikle terminalde `<< npm init >>` komutu çalıştırıldı ve gelen sorulara default olarak yanıt verildi. Ardından oluşan klasör içerisinde `<< package.json >>` adında yeni bir dosya oluştu. Az önce default olarak verilen yanıtlar bu dosya içerisinde mevcuttu.

Ardından kaynak kod çalıştırmada argüman gönderme işlemine bakıldı. Dosyamızın ismi `deneme.js` olma durumunda terminale `<< node deneme.js >>` yazma durumunda dosyayı çalıştırabiliyoruz. Artı olarak çalışma esnasında argüman da gönderebileceğimizi gördük. Bunu da şu şekilde sağlayabiliyoruz:

`<< console.log(process.argv) >>` kodunu terminalde argümanla çalıştırma durumunda (`node deneme.js Gizem`) dışarıdan alınan argüman çıktı olarak alınabilir.

```
PS C:\Users\Gizem\Desktop\ders2> node deneme.js Gizem
Debugger attached.
[
  'C:\\Program Files\\nodejs\\node.exe',
  'C:\\Users\\Gizem\\Desktop\\ders2\\deneme.js',
  'Gizem'
]
```

Burada `console.log` içerisine yazdığımız `argv` ise argüman vektörün kısaltışı anlamına gelmektedir.

Argüman olarak sadece bir tane göndermiş olmamıza rağmen üç çıktı aldık. Bunun ilki executable in konum bilgisi (node.exe). İkincisi çalıştırmak istediğimiz dosyanın konum bilgisi ve sonuncusu da girdiğimiz argümanın çıktısıdır. Argüman birden fazla girilebilir. Ama ne kadar girilirse girilsen her zaman bu sırada çıktı alırız.

Daha sonra da not ekleme ,silme ve listeleme işlemine bakıldı. Bunu yapmada işlemler dışarıdan argüman olarak alındı.

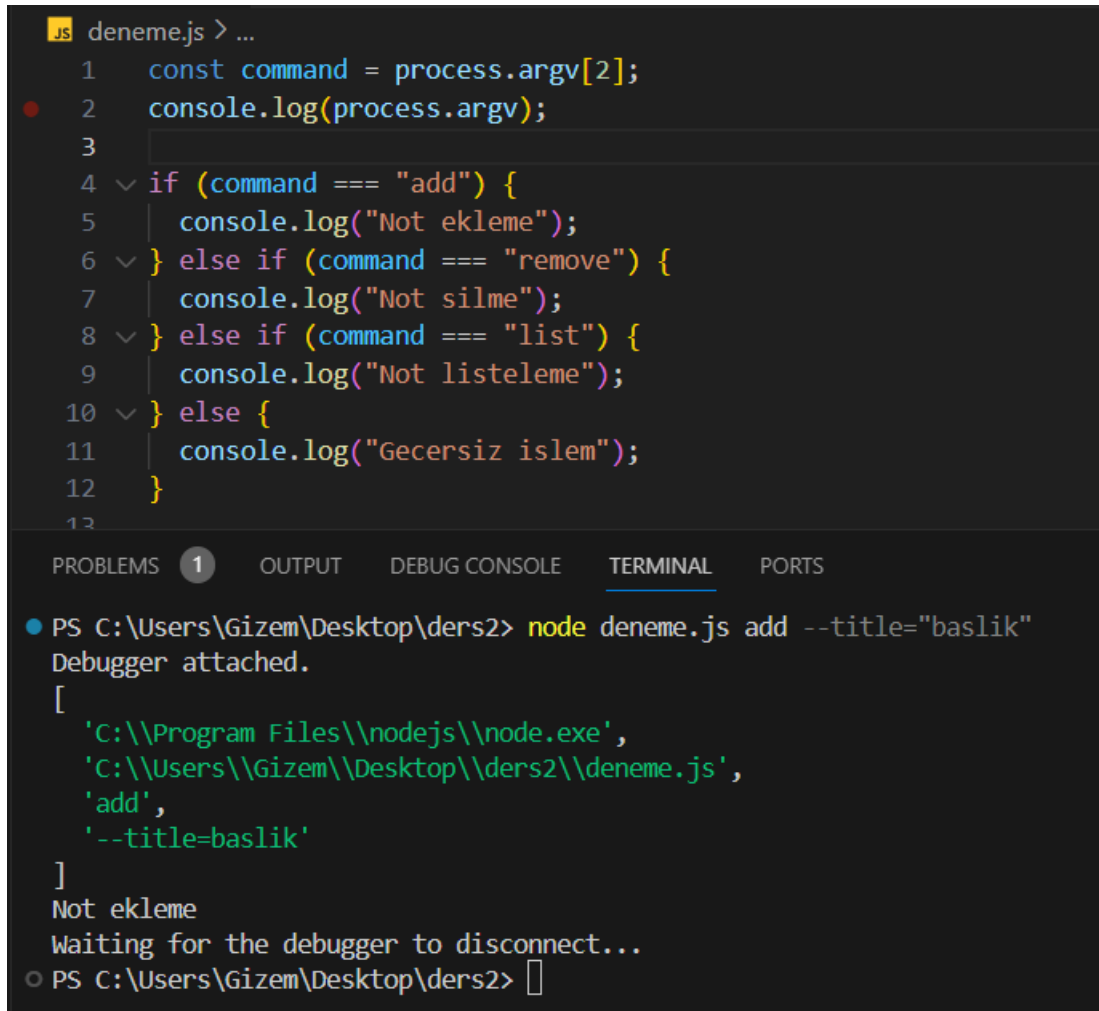
Az önce de bahsedildiği gibi ilk iki çıktı bizim işimize pek yaramayacağı için ikinci vektörden almak daha mantıklı oluyor ve bunu da şu kodla sağlayabiliyoruz.

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

Ardından dışarıdan eklenen değeri sorgulamak için if koşulu kullanıldı ve konsola çıktı alındı. Onu da bu kodla yaptık :

```
3   const command = process.argv[2]
4
5   if (command === "add"){
6       console.log("Not ekleme")
7   }
8   else if(command === "remove"){
9       console.log("Not silme")
10  }
11  else if(command === "list"){
12      console.log("Not listeleme")
13  }
14  else{
15      console.log("Gecersiz islem")
16  }
```

Ardından add komutu çalıştırıldı ve başlık bilgisi girildi. Başlık bilgisini terminalde girdik ve çıktı olarak da alabildik o da şu şekilde oldu:



```
JS deneme.js > ...
1  const command = process.argv[2];
2  console.log(process.argv);
3
4  if (command === "add") {
5      console.log("Not ekleme");
6  } else if (command === "remove") {
7      console.log("Not silme");
8  } else if (command === "list") {
9      console.log("Not listeleme");
10 } else {
11     console.log("Gecersiz islem");
12 }
13
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Gizem\Desktop\ders2> node deneme.js add --title="baslik"
Debugger attached.
[
  'C:\\Program Files\\nodejs\\node.exe',
  'C:\\Users\\Gizem\\Desktop\\ders2\\deneme.js',
  'add',
  '--title=baslik'
]
Not ekleme
Waiting for the debugger to disconnect...
PS C:\Users\Gizem\Desktop\ders2>
```

Ardından npm yargs modülünü kullanmayı gördük. Npm kullanmak için terminalde << npm i yargs>> kodunu çalıştırarak kodumuza aktarmayı sağladık.

Yargs modülünü kullanmak içinse gerekli bir kalıbımız bulunmaktadır. O kalıp ise:

```
Const yargs = require("yargs");
```

```

JS deneme1.js > ...
1  const yargs = require('yargs');
2  yargs.version ('1.1.0');
3  console.log(yargs.argv);
4
5  yargs.command({
6      command: 'add',
7      describe: 'yeni not ekler',
8      handler: function(argv){
9          console.log("yeni not ekleme");
10     }
11 })
12
13 yargs.command({
14     command: 'remove',
15     describe: 'secilen notu siler',
16     handler: function(){
17         console.log('not siliniyor');
18     }
19 })
20
21 yargs.command({
22     command: 'list',
23     describe: 'mevcut notu listeler',
24     handler: function(){
25         console.log('notlar listeleniyor');
26     }
27 })
28
29 yargs.command({
30     command: 'read',
31     describe: 'secilen notu gosterir',
32     handler: function(){
33         console.log('not gosteriliyor');
34     }
35 })
36
37 yargs.parse()

```

Yargs modülünde koşul kullanma gerektirmez.command ile girilen argümana bakılır. Describe ile açıklama sağlanır ve yapılacak işlem ise handler de belirlenir.

```
js app.js > ...
1  const yargs = require('yargs');
2  yargs.version ('1.1.0');
3  console.log(yargs.argv);
4
5  yargs.command({
6    command: 'add',
7    describe: 'yeni not ekler',
8    builder:{
9      title:{
10        describe: 'Not basligi',
11        demandOption: true,
12        type: 'string'
13      }
14    },
15    body:{
16      describe: 'Not icerigi',
17      demandOption: true,
18      type: 'string'
19    },
20    handler: function(argv){
21      console.log('baslik : ',argv.title);
22      console.log('icerik : ',argv.body);
23    }
24  })
25
```

```
26  yargs.command({
27    command: 'remove',
28    describe: 'secilen notu siler',
29    handler: function(){
30      console.log('not siliniyor');
31    }
32  })
33
34  yargs.command({
35    command: 'list',
36    describe: 'mevcut notu listeler',
37    handler: function(){
38      console.log('notlar listeleniyor');
39    }
40  })
41
42  yargs.command({
43    command: 'read',
44    describe: 'secilen notu gosterir',
45    handler: function(){
46      console.log('not gosteriliyor');
47    }
48  })
49
50  yargs.parse()
```

Bu kodda add komutunda builder komutunu ekledik ve içerisine yazdığımız bilgiyi anlamlandırdık. Bu sayede başlık atmama durumunda hata verdi yani başlık almayı zorunlu kılmış olduk.

Body bloğunda ise not ekleme kısmının türünü(string) ve eklemesinin gerekli olduğunu belirtmiş olduk.

Ardından File system modüllüne bakıldı. Onu da kod üzerinde anlatmak gerekirse yazmış olduğum kod:

```
js json_deneme.js > ...
1  const fs = require('fs')
2
3  const book = {
4    title: 'Kitap basligi',
5    author: 'Yazar',
6  }
7
8
9  const bookJSON=JSON.stringify(book)
10 fs.writeFileSync('denemeJson',bookJSON) //denemejson dosyası olusturuldu ve icerisine yazildi
11 console.log (bookJSON)
12 //console.log (bookJSON.title) //bunu tanimaz,erisemez
13
14 const parsedObject = JSON.parse(bookJSON)
15
16 const dataBuffer = fs.readFileSync('denemeJson') //dosyadan okuma islemi
17 console.log(dataBuffer.toString())
18 const dataJSON = dataBuffer.toString()
19 const data = JSON.parse(dataJSON)
20 console.log(data)
21 /*
22 console.log(parsedObject)
23 console.log(parsedObject.title) //artık json oldugundan oturu erisebiliriz
24 console.log(parsedObject.author) // aynı zamanda bu da erisilebilir
25 */
```

File System modülünü dosyaya tane require komutu ile ekledik

Kitabın bilgilerini json formatında bookJSON da tuttuk.

Ardından dosya oluşturup içerisine bilgileri aldık. Bunu da writeFileSync komutu ile yapabildik. BookJson üzerinden başlık veya yazar bilgisine erişemiyorduk onu da parsedObject ile sağladık.

Dosyadan okuma işlemini readFileSync komutu ile sağladık.