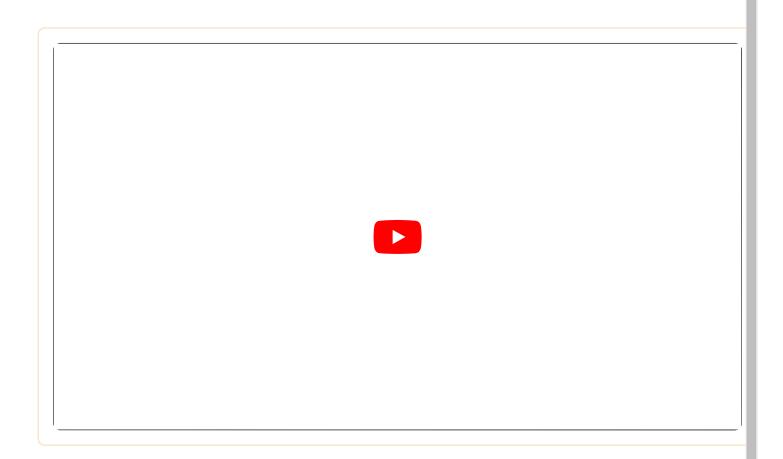
≔ ing Veri Türü İşlemleri



String Veri Türü İşlemleri

String nedir?

JavaScript metinlerin tümü "*string*" veri tipi içinde tutulur ve "*string*" veri türü ile ilgili yapabileceğimiz birden fazla işlem vardır. Bu yazımızda bunlardan bahsedeceğiz.

Length Özelliği - Uzunluk Değerini Alma

String veri tipinde bulunan bir ifadenin ne kadar uzun olduğunu bulmak için Length özelliğimizi kullanırız.

```
index.js x

index.js > ...

var isim = 'Oğuz';

console.log(isim.length);

3

C:\Users\oguz\Desktop\string-islemleri
\(\lambda\) node index.js

4
```

indexOf - Metin İçinde Arama Yapma

Metnin içinde aramak istediğimiz değerin <u>inde</u>x numarasını bize verir.

```
index.js X

index.js > var isim: string

var isim = 'Oğuz Duran';

console.log(isim.indexOf('Du'));
```

```
C:\Users\oguz\Desktop\string-islemleri
λ node index.js
5
```

lastIndexO f- Metin İçinde Arama Yapma

indexOf ile arasındaki tek fark aranan kelime birden fazla geçiyor ise en son eşleşmeden gelen index numarasını döndürür.

```
index.js X
index.js \ ...
1     var isim = 'Oğuz Duran';
2     console.log(isim.lastIndexOf('u'));

C:\Users\oguz\Desktop\string-islemleri
λ node index.js
6
```

Search - Metin İçinde Arama Yapma

indexOf ile aynı sonuçlara ulaşırız genel olarak "Regular Expressions" işlemleri için çok kullanılan bir metottur.

```
index.js ×
index.js > ...
var isim = 'Oğuz Duran';
console.log(isim.search('Du'));

C:\Users\oguz\Desktop\string-islemleri
node index.js
```

Slice - Metninden Parça Almak

Metin içinden almak istediğimiz yerlerin index numaralarını vererek metin içinde bulunan parçayı alabiliriz.

```
index.js x

index.js > ...

var isim = 'Oğuz Duran';

console.log(isim.slice(5, 10));
```

Bitiş index numarasındaki karakter sonuca dahil edilmez.

```
C:\Users\oguz\Desktop\string-islemleri
\( \lambda \) node index.js
Duran
```

Not: Tek index yazmak ise yazılan index numarasından sonra gelen tüm karakterleri almasına neden olur.

```
index.js ×

index.js >...

var isim = 'Oğuz Duran';

console.log(isim.slice(2));

c:\Users\oguz\Desktop\string-islemleri
λ node index.js

uz Duran
```

Replace – Metin Bulma ve Değiştirme

Aranan metni istediğimiz metin ile değiştirmemize olanak sağlar.

```
index.js X

index.js > ...

var isim = 'Oğuz Duran';

console.log(isim.replace('Duran', 'Oğuz'));

C:\Users\oguz\Desktop\string-islemleri
    \lambda node index.js
    Oğuz Oğuz
```

```
index.js X
index.js > ...

var isim = 'Oğuz Duran';

console.log(isim.toUpperCase());

console.log(isim.toLowerCase());
```

toUpperCase metin içinde bulunan tüm karakterleri büyük harf yapmamıza olanak sağlar.

toLowerCase metin içinde bulunan tüm karakterleri küçük harf yapmamıza olanak sağlar.

```
C:\Users\oguz\Desktop\string-islemleri
λ node index.js
OĞUZ DURAN
oğuz duran
```

Concat - Metin Birleştirme

Elimizde bulunan iki string türündeki veriyi birleştirmemize olanak sağlar.

```
index.js x

index.js > ...

var isim = 'Oğuz';

var soyad = 'Duran';

console.log(isim.concat(' ', soyad));

C:\Users\oguz\Desktop\string-islemleri
\[ \lambda \] node index.js
Oğuz Duran
```

charAt- İndex Numarasına Göre Karakter Bulmak

Belirtilen index numarasında yer alan karakteri verir.

```
index.js X

index.js X

index.js x

var isim = 'Oğuz Duran';

console.log(isim.charAt(1));

C:\Users\oguz\Desktop\string-islemleri

node index.js

ğ
```

charCodeAt – İndex Numarasına Göre Karakterin Unicode Değerini Bulma

Belirtilen index numarasında yer alan karakterin Unicode değerini verir.

```
index.js x

index.js > ...

var isim = 'Oğuz Duran';

console.log(isim.charCodeAt(1));

3

C:\Users\oguz\Desktop\string-islemleri
\(\lambda\) node index.js
287
```

Split - Metni Diziye Çevirme

Split metodu ile istenilen metin diziye çevrilebilir. Kullanılan parametre ile metnin nasıl parçalanacağı belirtilir.

```
index.js X

is index.js >...

1     var isim = 'Oğuz, Kamer';
2     var isimler = isim.split(',');
3     console.log(isimler);
4

C:\Users\oguz\Desktop\string-islemleri
\(\lambda\) node index.js
[ 'Oğuz', 'Kamer' ]
```

Bir String ifadesi içerisinde yer alan bilgilerin istediğimiz gibi olup olmadığını kontrol etmek isteyebiliriz. İçerisinde mail adresi geçiyor mu? Kullanıcın verilerinde benim aradığım bilgi var mı? Gibi kontroller yapmak istiyor olabiliriz. String veri türü işlemleri bize bu noktada yardımcı olur.

Örnekler üzerinden görelim:

```
let email = "kodluyoruz@kodluyoruz.org"
let firstName = "JavaScript"
let lastName = "BOOTCAMP"
```

Bu değişkenleri yazı boyunca kullanacağız.

Karakter Sayısı

String bir ifadenin karakter sayısını (uzunluğunu) bulmak istiyorsak length özelliğini kullanırız.

```
console.log(email.length); //25
console.log(firstName.length); //10
```

Karakter Yeri (Index)

String bir ifadenin içerisinde, aradığımız karakterin yerini bulmak için 2 yöntem vardır. [] ve chartAt() metodu bize bu konuda yardımcı olur.

```
console.log(firstName.[0]); //"J"
console.log(firstName.charAt(2)); //"v"
```

Büyük/Küçük Harfe Çevirme

String bir ifadeyi tamamen büyük veya küçük harf yapmak istiyorsak. Büyük harf için toUpperCase(), küçük harf için ise toLowerCase() metotlarını kullanırız.

```
firstName = firstName.toUpperCase();
console.log(firstName); //"JAVASCRIPT"

lastName = lastName.toLowerCase();
console.log(lastName); //"bootcamp"
```

İlk Harfi Büyük Kalan Harfleri Küçük Yapma

En başta değişkenlerimizi tanımlardan *lastName* değişkenimizi "BOOTCAMP" olarak sadece büyük harf ile yazdık. Bu ifademizi "Bootcamp" değiştirmek isteyebiliriz. Ya da sadece küçük harfler ile yazılmış bir özel ismin ilk harfini, büyük harf yapmak istiyor olabiliriz.

Bu tür durumları şu şekilde sağlarız.

```
lastName = `${lastName[0].toUpperCase()}${lastName.slice(1).toLowerCase()}` //Bootcamp
```

İstediğimiz Bilgiyi Aramak

String bir ifade içerisinde, istediğimiz bir bilginin yerini bulmak için search() metodunu kullanırız. Kontrol ettiğimiz ifade nereden başlıyor ise bize onun yerini verir.

```
console.log(email.search("@")); //10
```

Eğer arattığımız şey yok ise -1 sonucu verir.

```
console.log(email.search("olmayan")); //-1
```

search() yerine kullanabileceğimiz başka bir metot ise index0f() metodudur.

```
console.log(email.indexOf(".")); //21
```

Belli Bir Bilgiyi Almak

Biraz önce search() metoduyla @ işaretinin nerede olduğunu almıştık. İşte bu noktada @ işaretinden sonra gelen domain bilgisini almak bizim için çok kolay bir hale geldi, slice() metoduyla bu işlemi gerçekleştirebiliriz.

```
console.log(email.slice(10+1)); //"@kodluyoruz.org"
//@ ifadesini almak istemediğimiz için +1 kullandık.
```

Başka bir kullanımı ise şu şekildedir.

```
console.log(firstName.slice(1,4)); //"ava"
```

Yukarıdaki ifadede 1'inci index'ten başla ve 4. index'e kadar olan yeri al dedik. Bu noktada şunu anlamak önemli, 1. index'ten başlayıp 4. index'e kadar gittiğimiz için sadece 3 karakter aldı.

email değişkeninde sadece domaini almak istesek bunu iki farklı yöntemle yapabiliriz.

```
let domain = email.slice(email.search("@")+1);
console.log(domain); //"kodluyoruz.org"

let domain = email.slice(email.indexOf("@")+1);
console.log(domain); //"kodluyoruz.org"
```

Bilgiyi Değiştirmek

email değişkeninde domaini değiştirmek istersek eğer, replace() metodu bize yardımcı olur.

```
email = email.replace("kodluyoruz.org", "gmail.com");
console.log(email); //"kodluyoruz@gmail.com"
```

İstediğim Bilgi Var Mı?

Aradığımız bilginin değişken içerisinde olup olmadığını kontrol etmek için includes() metodunu kullanırız. true veya false döner.

```
email.includes("@"); //true
email.includes("$"); //false
```

Nasıl Başladı Nasıl Bitti?

Aldığımız veri istediğimiz bilgiyle başladı mı? Veya istediğimiz bilgiyle bitti mi? Bu durumu kontrol etmek için startsWith() ve endsWith() metodunu kullanırız.

```
email.startsWith("@"); //false
email.endsWith("org"); //true
```

Sorular

```
let url = "www.kodluyoruz.org";
let language = "Java";
```

Soru 1: Yukarıdaki language değişkenin değerini JavaScript olarak değiştirin ve konsola yazdırın.

Çözüm Soru 1

Soru 2: Yukarıdaki *url* değişkenini kullanarak "www" olmadan *domain* (kodluyoruz.org) adında yeni bir değişken oluşturun ve konsola yazdırın.

Çözüm Soru 2

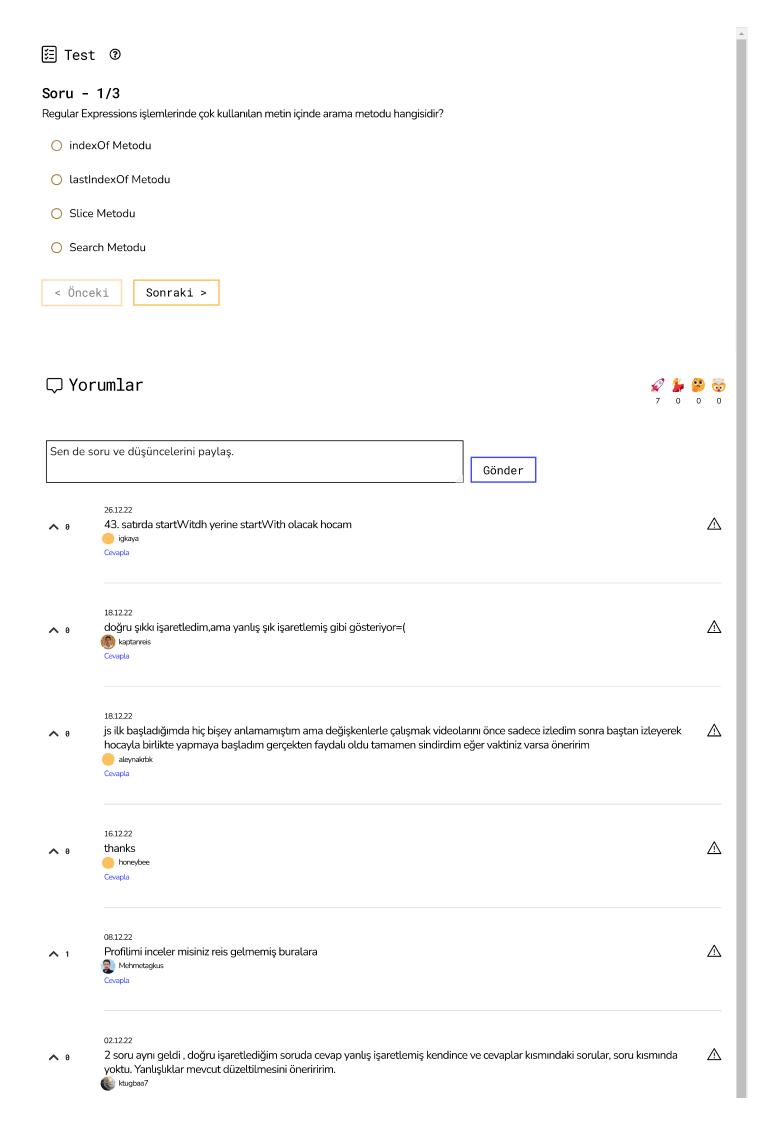
Kaynaklar

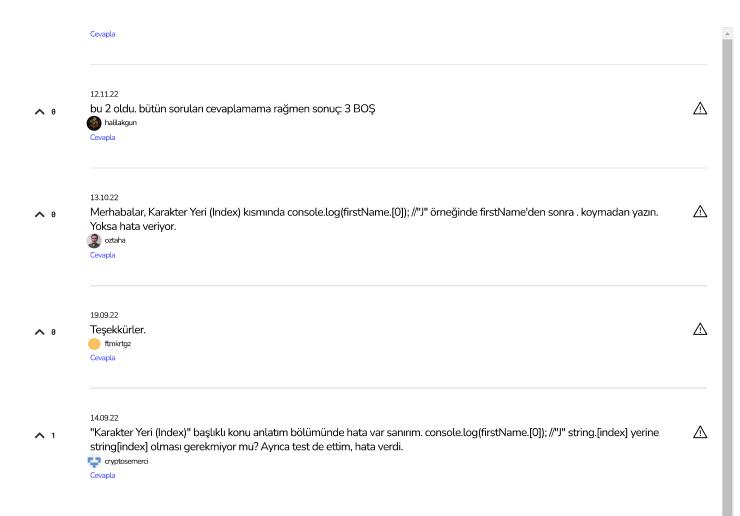
• JavaScript String Metotları

String Örnekleri

String yapısını öğrenelim.







3179 Members Online

