

STRING AND STRING METHODS

concat () : İki veya daha fazla dizeyi birleştirmek için kullanılır. Bu yöntem var olan dizeleri değiştirmez, ancak birleştirilmiş dizelerin metnini içeren yeni bir dize döndürür.

```
let x = "Hello my friend.";
let y = "How are you?";
console.log(x.concat(y));
"Hello my friend.How are you?"
```

endsWith () : Dizenin belirtilen karakterler ile bitip bitmediğini kontrol eder. Bitiyorsa true, bitmiyorsa false döndürür.

```
let x = "Hello my friend";
console.log(x.endsWith("friend")); true
console.log(x.endsWith("Hello")); false
```

startsWith () : Dizenin belirtilen karakterler ile başlayıp başlamadığını kontrol eder. Başlıyorsa true, başlamıyorsa false döndürür.

```
let x = "Hello my friend";
console.log(x.startsWith("friend")); false
console.log(x.startsWith("Hello")); true
```

includes () : Dizenin belirtilen karakterleri içerip içermediğini kontrol eder. İçeriyorsa true, içermiyorsa false döndürür.

```
let x = "Hello my friend";
console.log(x.includes("my")); true
console.log(x.includes("hi")); false
```

indexOf () : Dizede belirtilen karakterin ilk oluştuğu yerin konumunu döndürür.

```
let x = "Hello my friend";
console.log(x.indexOf("my")); 6
```

lastIndexOf () : Dizede belirtilen bir karakterin son oluştuğu konumu döndürür.

```
let x = "Hello my friend. Can I take my books?";
console.log(x.lastIndexOf("my")); 28
```

padEnd () : Dizeyi belirtilen karakterlerle belirtilen indexe kadar doldurur. Dizenin sonundan başlar.

```
let x = "Hello my friend.";
console.log(x.padEnd(40, "."));
"Hello my friend....."
```

padStart () : Dizeyi belirtilen karakterlerle belirtilen uzunluğa kadar doldurur. Dizenin başından başlar.

```
let x = "Hello my friend.";
console.log(x.padStart(40, "."));
".....Hello my friend."
```

slice () : Bir dizenin belirtilen bölümlerini ayıklar ve çıkarılan parçaları yeni bir dizede döndürür.

```
let x = "Hello my friend.";
console.log(x.slice(5, 15));
"my friend"
```

split () : Bir dizeyi belirtilen karakterlerinden bir alt dizeler dizisine bölmek için kullanılır ve yeni diziyi döndürür.

```
let x = "Hello my friend.";
console.log(x.split("e"));
[ 'H', 'llo my fri', 'nd.' ]
```

substring () : Dizenin belirtilen başlangıç ve bitiş index kısmını döndürür. → (from, to)

```
let x = "Hello my friend.";
console.log(x.substring(2, 4));
"ll"
```

substr () : Dizeyi belirtilen index ile başlayıp belli bir uzunluğa kadar döndürür. → (from, length)

```
let x = "Hello my friend.";
console.log(x.substr(2, 4));
"llo"
```

trim () : Dizenin başındaki ve sonundaki boşlukları kaldırır.

```
let x = "    Hello my friend.    ";  
  
console.log(x.trim());  
  
"Hello my friend."
```

trimEnd () : Dizenin sonundaki boşluğu kaldırır.

```
let x = "    Hello my friend.    ";  
  
console.log(x.trimEnd());  
  
"    Hello my friend."
```

trimStart () : Dizenin başındaki boşluğu kaldırır.

```
let x = "    Hello my friend.    ";  
  
console.log(x.trimStart());  
  
"Hello my friend.    "
```

repeat () : Dizeyi belirtilen sayıda tekrar ederek döndürür.

```
let x = "Hello my friend.";  
  
console.log(x.repeat(3));  
  
"Hello my friend.Hello my friend.Hello my friend."
```

replace () : Dizenin belirtilen bir karakterlerini başka bir belirtilen karakter ile değiştirerek döndürür.

```
let x = "Hello my friend.";  
  
console.log(x.replace("Hello", "Hi"));  
  
"Hi my friend."
```