

Tartalom

String, karakterlánc készítése	2
charAt()	2
concat()	2
includes()	2
indexOf()	2
length	3
repeat()	3
replace()	3
replaceAll()	3
substr()	3
substring()	4
slice()	4
split()	4
toLowerCase()	5
toUpperCase()	5
trim(), trimStart() & trimEnd()	5
Hivatalos Források/leírások	6

String, karakterlánc készítése

Alapvetően minden szövegszerkesztéshez szükség van egy alap stringre ezt a következő módon hozom létre.

```
let egyszeruSzoveg="Ez egy szöveg";
```

charAt()

Kiválaszt egy adott indexű karaktert a szövegből és annak az értékét visszaadja. (0-tól történik az indexelés)

```
let kivalasztottKarakter=egyszeruSzoveg.charAt(1);  
document.write(kivalasztottKarakter); //egy karakter értékével tér vissza
```

concat()

Hozzáfűz egy másik stringet az eredeti szöveghez.

```
let kiegészítettSzoveg=egyszeruSzoveg.concat(", amihez hozzáírtam egy másik szövegrészt!");  
document.write(kiegészítettSzoveg); //Az eredeti szöveghez hozzáfűzi a concatben lévő értéket!
```

includes()

Keresünk egy adott karakterláncot a szövegben, true, vagy false értékkel térhet vissza.

```
let kereses1=egyszeruSzoveg.includes("egy");  
let kereses2=egyszeruSzoveg.includes("kettő");  
document.write(kereses1); //logikai igaz értékkel tér vissza  
document.write(kereses2); //logikai hamis értékkel tér vissza
```

indexOf()

Adott szöveg kezdeti karakterének indexét.

```
let keresettElemIndexe1=egyszeruSzoveg.indexOf("egy");  
let keresettElemIndexe2=egyszeruSzoveg.indexOf("ketto");  
document.write(keresettElemIndexe1); //3 értékkel tér vissza mert ott van az egy szó kezdő indexe  
document.write(keresettElemIndexe2); //-1 értékkel tér vissza mert nincs a keresett rész a szövegben
```

length

Adott szöveg hosszát adja vissza.

```
let szoveghossz=egyszeruSzoveg.length;  
document.write(szoveghossz); //szám típusú értékkel tér vissza, ami a szöveg hossza
```

repeat()

Ismétli az adott szöveget, a paraméterként megadott mennyiségben.

```
let ismeteltSzoveg=egyszeruSzoveg.repeat(3);  
document.write(ismeteltSzoveg); //ismétli a szöveget 3-szor(az aktuális esetben)
```

replace()

Kicserél egy adott szöveget egy új szövegre (első paramétert--> második paraméterre). Egy alkalommal, több lehetőség esetén az első helyen lévő cseréli ki.

```
let kicsereltSzoveg=egyszeruSzoveg.replace("szöveg","mondat");  
document.write(kicsereltSzoveg); //”Ez egy mondat” karakterlánccal tér vissza
```

replaceAll()

Kicserél egy adott szöveget egy új szövegre (első paramétert--> második paraméterre). Az összes lehetséges előfordulási helyen.

```
let kicsereltSzoveg=egyszeruSzoveg.replaceAll("szöveg","mondat");  
document.write(kicsereltSzoveg); //”Ez egy mondat” karakterlánccal tér vissza
```

substr()

Kivág egy adott szöveget, a kezdőkarakter indexét és a kivágni kívánt szöveg hosszát megadva, azok között.

```
let kivagottSzoveg=egyszeruSzoveg.substr(7,6);  
document.write(kivagottSzoveg); //”szöveg” karakterlánccal tér vissza
```

substring()

Kivág egy adott szöveget, a kezdőkarakter indexét és az utolsó karakter indexét megadva, azok között.

```
let substringSzoveg=egyszeruSzoveg.substring(7,13);  
document.write(substringSzoveg); //”szöveg” karakterlánccal tér vissza
```

slice()

Kivág egy adott részt a szövegből, kezdő és végső indexet megadva

```
let sliceSzoveg1=egyszeruSzoveg.slice(3,6);  
document.write(sliceSzoveg1); //”egy” karakterlánccal tér vissza
```

Negatív érték esetén a karakterlánc végén lévő szövegrészt vág ki, hasonlóan, mint pozitív esetén.

```
let sliceSzoveg1=egyszeruSzoveg.slice(-6);  
document.write(sliceSzoveg1); //”szöveg” karakterlánccal tér vissza
```

```
let sliceSzoveg3=egyszeruSzoveg.slice(7,13);  
document.write(sliceSzoveg3); //”szöveg” karakterlánccal tér vissza
```

split()

Földarabolja az adott szöveget, egy elválasztó karakter alapján!

```
var darabolandoSzoveg="alma,körte,barack,szilva";  
var daraboltSzoveg=new Array(darabolandoSzoveg.split(','));  
for(let i=0;i<daraboltSzoveg.length;i++){  
    document.write(daraboltSzoveg[i]);  
}  
document.write(daraboltSzoveg); //vagy így is kiírathatom a tömbömet  
const daraboltSzoveg2=darabolandoSzoveg.split(',');  
for(let i=0;i<daraboltSzoveg2.length;i++){  
    document.write(daraboltSzoveg2[i]);  
}
```

join()

Összefűzi a tömb elemeit karakterlánccá (splittelés ellentéte!)

```
var darabolandoSzoveg=["alma","körte","barack","szilva"];  
let stringTomb= darabolandoSzoveg.join();  
document.write(stringTomb); //”alma,körte,barack,szilva ” karakterlánccal tér vissza
```

toLowerCase()

Kisbetűssé alakítja a szöveget.

```
let kisbetus=egyszeruSzoveg.toLowerCase();  
document.write(kisbetus); //”ez egy szöveg” karakterlánccal tér vissza
```

toUpperCase()

Nagybetűssé alakítja a szöveget.

```
let nagybetus=egyszeruSzoveg.toUpperCase();  
document.write(nagybetus); //”EZ EGY SZÖVEG” karakterlánccal tér vissza  
//Capitalize nincs!!!
```

trim(), trimStart() & trimEnd()

Levágja a fölösleges szóközöket a string széleiről.

```
var szovegSokSzokozzel="    Sok felesleges szóköz van ittene!    ";  
let simaTrim=szovegSokSzokozzel.trim();  
let balTrim=szovegSokSzokozzel.trimStart();  
let jobbTrim=szovegSokSzokozzel.trimEnd();  
document.write(simaTrim); //String szélein lévő felesleges szóközöket levágja  
document.write(balTrim); //String elején lévő felesleges szóközöket levágja  
document.write(jobbTrim); //String végén lévő felesleges szóközöket levágja
```

Hivatalos Források/leírások

- charAt()
- concat()
- includes()
- indexOf()
- length
- repeat()
- replace()
- substr()
- substring()
- split()
- slice()
- toLowerCase()
- toUpperCase()
- trim()
- trimStart()
- trimEnd()