

Tartalom

Miről lesz itt szó?	2
Tömbök	2
Deklarálás és értékadás	2
Hivatkozás adott elemre	2
Tömbök bejárása DOM felületen	2
Objektumok használta Angularban	3
Interface létrehozás	3
Objektum létrehozása interface segítségével	3
Elemekre való hivatkozás	3
Objektum típusú tömbök	4
Deklarálás és értékadás	4
Adott objektum elem értékeinek kiírása	4
Objektum típusú tömb bejárása DOM felületen	4
Lista generálása: frontend kód	5
Weboldalon való megjelenés(részlet)	5
Táblázat generálása: frontend kód	5
Weboldalon való megjelenés (részlet):	6
További források	7

Miről lesz itt szó?

Az Angularban, mint korábban a JavaScriptben is tudunk összetett adatszerkezeteket kezelni. A következő dokumentumban, azt fogom bemutatni, mik ezek az adatszerkezetek, ezekre hogyan hivatkozhatunk, és hogyan jeleníthetjük meg azokat, többféle módon is.

Tömbök

Deklarálás és értékadás

Létrehozásuk a TypeScript szabályainak felel meg

```
bevasarloLista: string[] = ["tej", "kenyér", "sör", "alma", "csokis  
tej"]
```

A név megadása után meg kell adni annak típusát, illetve jelezni kell, hogy tömböt akarunk generálni, majd szögletes zárójelek között fel lehet sorolni az elemeket vesszővel elválasztva...

Hivatkozás adott elemre

Az elemekre ugyanakkor külön-külön is hivatkozhatunk, a korábban tanult, tömbNeve[index] típusú hivatkozással.

Tömbök bejárása DOM felületen

Az elemek listázására egy speciális Angular elemet az ***ngFor**-t használjuk. Ami leginkább a JavaScript **foreach** ciklusára hasonlít.

```
<ul>  
  <li *ngFor="let termék of bevasarloLista">  
    {{termék}}  
  </li>  
</ul>
```

Az ***ngFor** után egyenlőség jelet téve az alábbi paramétereket kell megadni

let változoNeve – Utal arra, milyen néven tudunk hivatkozni, az elemekre

of AdatszerkezetNeve – Megadja melyik az az adatszerkezet, amit bejárunk a segítségével

Objektumok használata Angularban

Objektumok használata előtt érdemes interface-t is használnunk, melyek segítik a hibamentes létrehozást, azzal, hogy egy vázat generálnak, a „leendő” adatszerkezetünknek, ezeket az fő osztályon (class-on) kívül kell létrehozni. És beexportálni a projektbe, például.

Interface létrehozás

```
export interface versenyEredmeny {  
  helyezes: number;  
  meret: number;  
  sportag: string;  
  versenyszam: string;  
}
```

Objektum létrehozása interface segítségével

A korábban létrehozott interface-t immáron használhatjuk az objektum példány létrehozására.

```
interfaceElem:versenyEredmeny={  
  helyezes: 1,  
  meret:6,  
  sportag:"labdarúgás",  
  versenyszam:"kispályás football"  
}
```

Ilyenkor a TypeScript ügyel arra, hogy minden elemet létrehozzuk, amit korábban megadtunk!

Elemekre való hivatkozás

Ha kész az elem, a korábban már megtanult hivatkozás, **objektumNév.tulajdonságNév** segítségével hivatkozhatunk a létrehozott elemre.

```
<p>{{interfaceElem.helyezes}} helyezés {{interfaceElem.meret}} fő  
{{interfaceElem.sportag}} sportágban,  
  {{interfaceElem.versenyszam}} versenyszámban</p>
```

Objektum típusú tömbök

Deklarálás és értékadás

Természetesen nem csak egyszerű objektumokat, hanem objektum típusú tömböket is könnyedén létrehozhatunk. Szintén interface segítségével, mely segít a pontos értékek típusának vizsgálatában.

```
ObjektumFeltolto(feltoltendoElem: string[]): versenyEredmeny[] {}
```

A létrehozáskor ügyeljünk a megfelelő típus megadására, ebben az esetben **versenyEredmeny** és a **[]** szimbólumokra, mellyel jelezzük, hogy egy tömböt szeretnénk létrehozni, ami Verseny eredmény típusú objektumokat tartalmaz.

```
megjelenitendoAdatok: versenyEredmeny[] =  
this.ObjektumFeltolto(this.eredmenyek);
```

Adott objektum elem értékeinek kiírása

Természetesen adott elem értékeire is hivatkozhatunk, ilyenkor a tömbben elfoglalt indexe mellett meg kell adni a tulajdonságot is, amit szeretnénk megjeleníteni/használni.

Objektum típusú tömb bejárása DOM felületen

Mivel szintén összetett adatszerkezetről beszélünk, a ***ngFor** parancs ismét használható, a DOM részben. A paraméterezés hasonló, mint a sima tömbök esetében.

Az ***ngFor** után egyenlőség jelet téve az alábbi paramétereket kell megadni

let valtozoNeve – Utal arra, milyen néven tudunk hivatkozni, az elemekre

of AdatszerkezetNeve – Megadja melyik az az adatszerkezet, amit bejárunk a segítségével

Interface típusú tömb esetén a **valtozoNeve.tulajdonsagNeve** értékkel hivatkozunk az adott elemekre!

Lista generálása: frontend kód

```
<ul>
  <li *ngFor="let objektumAdat of megjelenitendoAdatok">
    {{objektumAdat.helyezes}}. helyezés
    ({{objektumAdat.meret}}) fő
    [{{objektumAdat.sportag}}] -
    {{objektumAdat.versenyszam}}]
  </li>
</ul>
```

Weboldalon való megjelenés(részlet)



A screenshot of a web page displaying a list of sports results. The list is contained within a light gray box with a thin border. It shows seven entries, each consisting of a rank, a name in Hungarian, and a sport name in brackets. The text is slightly blurred, but the structure is clear.

- 1. helyezés (1) fő [atletika - kalapacsvetes]
- 1. helyezés (1) fő [uszás - 400m_gyorsuszas]
- 1. helyezés (1) fő [birkozas - kotott_fogas_legsuly]
- 1. helyezés (1) fő [torna - talajtorna]
- 1. helyezés (1) fő [torna - felemas_korlat]
- 1. helyezés (1) fő [vivas - kardvivas_egyen]
- 1. helyezés (1) fő [boklivas - nagypalacska]

Táblázat generálása: frontend kód

```
<h2>Táblázat adatainak megjelenítése</h2>
<table class="table table-striped table-dark table-sm">
  <thead>
    <tr>
      <th>Helyezés</th>
      <th>Csapat méret</th>
      <th>Sportág</th>
      <th>Versenyszám</th>
    </tr>
  </thead>
  <tbody>
    <tr *ngFor="let objektumAdat of szurtElemek">
      <td>{{objektumAdat.helyezes}}</td>
      <td>{{objektumAdat.meret}}</td>
      <td>{{objektumAdat.sportag}}</td>
      <td>{{objektumAdat.versenyszam}}</td>
    </tr>
  </tbody>
</table>
```

Weboldalon való megjelenés (részlet):
Táblázat adatainak megjelenítése

Helyezés	Csapat méret	Sportág	Versenyszám
1	1	atletika	kalapacsvetes
1	1	uszas	400m_gyorsuszas
1	1	birkozas	kotott_fogas_legsuly
1	1	torna	talajtorna
1	1	torna	felemas_korlat
1	1	vivas	kardvivas_egyen
1	1	okolvivas	nagyvaltosuly

További források

<https://v17.angular.io/api/common/NgFor>

<https://v17.angular.io/tutorial/first-app/first-app-lesson-04>

<https://v17.angular.io/tutorial/first-app/first-app-lesson-08>

<https://medium.com/@manikandanthangaraj/the-importance-of-interfaces-in-angular-d5142e998e0a>

<https://medium.com/@abdoessamadhmayda/angular-ngfor-complete-guide-b9d002144532>