

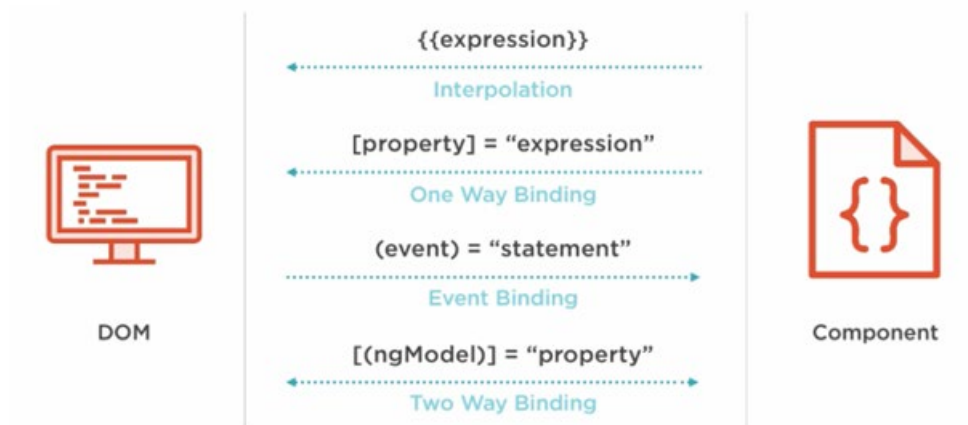
Tartalom

Mi is az a Data Binding?	2
Adat kötés típusai	2
Interpolation – Az egyszerű hivatkozás	2
Backend kód:	2
Frontend kód:	2
One Way vagy Property binding – A tulajdonság hivatkozás	3
Backend kód:	3
Frontend kód:	3
Event Binding – Események kezelése	3
Backend kód: Hivatkozott esemény a komponensben	3
Frontend kód: Eseményre hivatkozás a DOM-ban	3
Two Way Binding – A kétirányú kötés	4
Frontend kód:	4
Backend kód:	4
Weboldalon való megjelenés:	4
Űrlap elemek használata	5
Felhasznált források	6

Mi is az a Data Binding?

Az Angular adatbinding (data binding) rendszere lehetővé teszi az adatok szinkronizálását a komponensek és a DOM (Document Object Model) között. Az adatbinding különböző módjai segítenek a fejlesztőknek egyszerűbben és hatékonyabban kezelni az adatokat és az eseményeket az Angular alkalmazásokban.

Az Angular négy fő adatkötési módszert kínál: interpoláció, property binding, event binding és kétirányú adatkötés.



Adat kötés típusai

Interpolation – Az egyszerű hivatkozás

Szerkezete: `{{expression}}` magyarul `{{kifejezés}}`

Mint a mellékelt leírásban is látható, ilyenkor a komponensben található elem értékét jelenítjük meg a DOM felületén, elkészíteni igen egyszerű, az komponensben lévő elemre való hivatkozást dupla kapcsoszárojelek közé helyezzünk, az alábbi módon.

Backend kód:

```
szoveg: string = "Szeretem a programozást"
```

Frontend kód:

```
<p>{{szoveg}}</p>
```

One Way vagy Property binding – A tulajdonság hivatkozás

Szerkezete: `[property] = "expression"` magyarul `[tulajdonság]="kifejezés"`

A kommunikáció hasonló, mint az előbb, a komponensben lévő értéket jelenítjük meg a DOM felületén, hivatkozásakor kapcsos zárójelet használunk, ezzel utalunk rá, hogy ott egy ilyen hivatkozás lesz majd, utána pedig egyenlőségjellel megadjuk az értéket tartalmazó változó nevét.

Backend kód:

```
kepLink: string = "https://cyber-suli/kep.png";
kepCel: string = "_blank";
kepSzelesseg: number = 200;

kepUtvonal: string = "https://www.youtube.com/CybersProgTech
kepCimke: string = "Kövess be!";
kepAlternativSzoveg = "Cyber's Prog tech youtube csatorna"
```

Frontend kód:

```
<a [href]="kepUtvonal" [target]="kepCel"><img [src]="kepLink"
[alt]="kepAlternativSzoveg" [title]="kepCimke"
[width]="kepSzelesseg"></a>
```

Event Binding – Események kezelése

Szerkezete: `(event) = "statement"` magyarul `(eseménytípus) = "cselekmény"`

Itt fordított az irány az előzőkhez képest, a DOM eseményre reagálva hajtunk végre valamit a komponensen, ilyen esemény lehet egy egyszerű klikk, mint a mintában, vagy más egér mozgással kapcsolatos esetleg billentyűzet esemény is.

Backend kód: Hivatkozott esemény a komponensben

```
uzenet(): void {
  alert("Üdv az oldalamon!");
}
```

Frontend kód: Eseményre hivatkozás a DOM-ban

```
<button type="button" class="btn btn-primary"
(click)="uzenet()">Csapat infó</button>
```

Two Way Binding – A kétirányú kötés

Végül de nem utolsó sorban a két irányú kötés, amely talán az egyik leglátványosabb funkció, melyet idáig csak komplex JavaScript kóddal lehetett megvalósítani. Ilyenkor a DOM-ban módosított érték módosul a komponensben, és a komponens módosítások azonnal megjelennek a DOM-ban is viszont.

Ehhez nem kell mást tennünk, mint annak az elemnek az **[(ngModel)]** tulajdonságot megadni, majd egyenlőségjel után a nevét, amit szeretnénk, hogy felvegyen a komponensben.

Frontend kód:

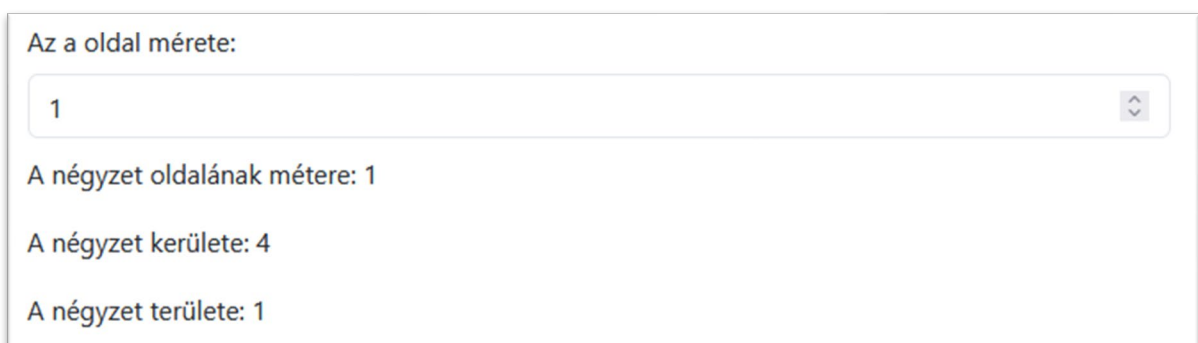
```
<label for="aOldal">Az a oldal mérete:</label>
<input type="number" class="form-control w-100 my-2" id="aOldal"
[(ngModel)]="aOldal">
<p>A négyzet oldalának métere: {{aOldal}}</p>
<p>A négyzet kerülete: {{aOldal*4}}</p>
<p>A négyzet területe: {{aOldal*aOldal}}</p>
```

Backend kód:

```
aOldal: number = 1;
```

A képen jól látjuk, hogy az input mezőt [(ngModel)] értékkel láttuk el, így, ha a benne lévő érték (aOldal) módosul, akkor a komponensben lévő ezzel azonos nevű elemet is módosítja, és a komponens módosítások, azonnal megjelennek azokon a helyeken, ahol arra az elemre hivatkozunk, a DOM-ban!

Weboldalon való megjelenés:



Az a oldal mérete:

1

A négyzet oldalának métere: 1

A négyzet kerülete: 4

A négyzet területe: 1

Űrlap elemek használata

Űrlap elemek használata esetén szükségünk van a forms modul importálására is. Melyet az **app.module.ts** állományban meg kell hívnünk a következő 2 sor importálásával, az elejére a következő kell...

```
import { FormsModule } from '@angular/forms';
```

illetve az import részben is fel kell sorolnunk az alábbi módon:

```
imports: [  
  BrowserModule,  
  AppRoutingModule,  
  FormsModule  
],
```

Ha ezeket importáljuk, már nem fog hibákat dobálni, az űrlap elemek használata esetén.

Felhasznált források

<https://v17.angular.io/guide/binding-syntax>

[https://en.wikipedia.org/wiki/Angular_\(web_framework\)](https://en.wikipedia.org/wiki/Angular_(web_framework))

<https://v17.angular.io/guide/interpolation>

<https://v17.angular.io/guide/property-binding>

<https://angular.dev/guide/templates/event-binding>

<https://angular.dev/guide/templates/two-way-binding>