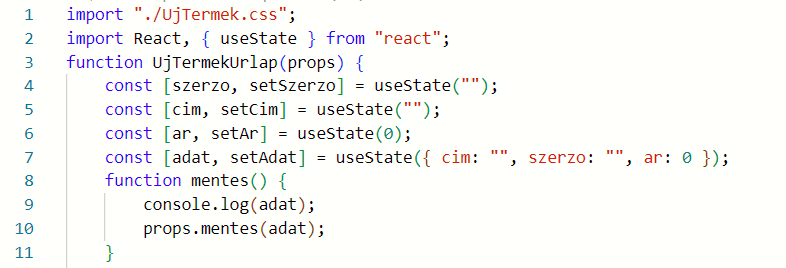
## Űrlapok React-ben

A formok olyan szempontból különlegesnek számítanak a Reactban, hogy minden egyes űrlap saját belső állapottal rendelkezik. A belső állapotot érdemes a React-ban a form saját Stat-jén keresztül kezelni.

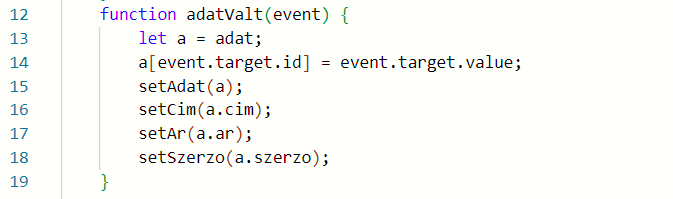
Azokat a Form elemeket, melyeket a react kontrollál, **Kontrollált komponensnek** nevezzük. Ezeket tehát a react nem csak rendereli, hanem irányítja is.

Egy HTML oldal esetében a Formot a DOM kezeli. React-ben a form adatait a componens kezeli. Az összes adat a a componens state-jében tárolódik.

Amennyiben a form elemeknek kezdőértéket akarunk adni, adjuk meg a state értékeit a függvénykomponens elején, és a formon a value tulajdonságnak adjuk értékül a state megfelelő értékét.



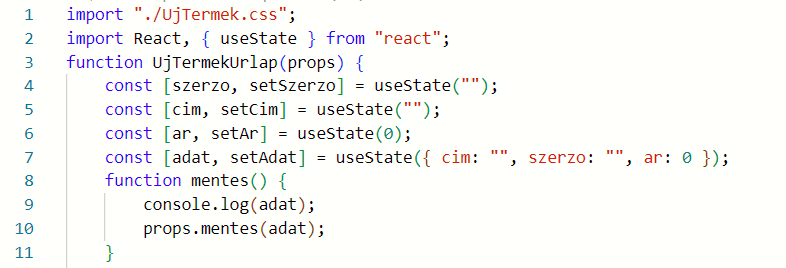
Az adatok kezeléséhez az input elemek onChange eseménykezelőjét kell használni. Az eseménykezelőben a state megfelelő értékét kell frissíteni. A state frissítésekor figyeljünk arra, hogy előbb az adatról készítsünk másolatot ( let a= adat), majd az másolt adaton frissítsük az értékeket, és a setFüggvényeknek ezzel a másolt adattal adjuk át az értékeket.





### Form adatainak küldése

1. Helyezzünk el egy submit gombot a form tagben.
2. FIGYELEM! **onSubmit**  eseménykezelőt a **form** elemre kell tenni, *nem a submit inputra*!
3. Írjuk meg az **eseménykezelő** függvényünket: meghívjuk benne a szülőelemünk props-on keresztül kapott függvényét.





### ÖSZSEFOGLALVA: Mi történik a komponensben?

1. Amikor először megjelenik a DOM-ban, az inputok értéke feltöltődik a **state**-ből.
2. Amikor a felhasználó beír valamit az inputba, az **onChange** eseménykezelő által meghívott függvényben a setState() frissíti a megfelelő értéket. Az input értéke ugyanaz marad, hiszen nincs minek változni.
3. Amikor a felhasználó elküldi a formot a form **onSubmit** eseménykezelőjében definiált függvény hívódik meg. Például itt gyűjthetjük össze és készíthetjük elő az adatokat a továbbküldéshez, megtörténhet az adatok ellenőrzése, stb. Majd a state kiürül, ami triggerel egy változást mind az inputok értékében, mind az „Output” oldalon.

### Textarea

REACT-ban a Textarea elem értékének beállításához is a value attribútumot használjuk. Innentől kezdve a kezelése az input elemhez hasonlóan történik.

### SELECT

A selecet elemben a html-ben a kiválasztott elemhez a selected attribútumot használtuk a kiválasztott option elemen. REACT e helyett itt is a szülőelemre (select) elhelyezi a value attribútumot

### A Fájlbeviteli címke

Kontrollálatlan komponens, mivel csak olvasható bementtel rendelkezik. <input type="file">

### Több input bemenet kezelése

Több input bemenet kezeléséhez használjuk a value helyett a name attribútumot. Az eseménykezelő az evenet.target.name alapján fogja tudni, hogy mit kell tenni:

## Feladat

Készítsd el az alábbi űrlapot React segítségével:



Amennyiben valamelyik elem nincs kitöltve, írj üzenetet, a hibákat jelezd a visszajelző űrlapon piros színnel.

Hogy lehetne megoldani, hogy a ki nem töltött űrlapmezőt is kiszínezzük?

Forrás: <https://hu.reactjs.org/docs/forms.html>

<https://hu.reactjs.org/docs/conditional-rendering.html>