מפת הדרכים של ריאקט:

1. אינטראקציה (כפתורים, inputs...)
2. תנאים להגיע לזה (רקע ב JS):
   * Array Destructuring
   * Events
3. **Hooks**

Hooks

מה זה hook? דרך לשמור משתנים בתוך הקומפוננטה גם אחרי שהיא מתרפרשת.

ריאקט אחראי לרפרש את הקומפוננטות.

נניח שאנחנו רוצים להציג מספר על המסך ולהגדיל אותו באחד כשלוחצים על כפתור.

הבעיה – ריאקט מאפס חלק מהמשתנים בצורה לא צפויה. לכן אי אפשר להגדיר סתם משתנה בקומפוננטה ולצפות שהערך ישמר.

לבעיה הזאת קוראים בעולם התכנות STATE

הפתרון של ריאקט – hooks

Hook זאת פונקציה מובנית של react בשביל להגדיר משתנים בצורה בטוחה.

useState

המטרה של userState הוא פשוט להגדיר משתנה כלומר לעשות השמה.

לסיכום: במקום השמה נשתמש ב useState

תרגיל כיתה:

1. לשפר את המונה כך שיהיה מוגבל ל 10 קליקים. אחרי 10 קליקים לא עולה יותר המספר.
2. בנוסף שירשום אחרי 10 קליקים "too many clicks"
3. בנוסף לקבל את המקסימום קליקים ב props
4. במקום ב props לקבל את המקסימום ב Input (2 hooks אחד למקסימום והשני למונה)
5. להוסיף גם input שמגדיר מינימום קליקים שחייבים לעשות. כל עוד לא עשית מספיק רושם

"not enough clicks"

אפשר להגדיר יותר מ hook אחד, עבור כל משתנה שנרצה לשמר.

לסיכום:

* קומפוננטות
* Props
* Events
* Map, filter
* Hooks

הדבר האחרון: שמירת נתונים בתוך הדפדפן כלומר עבודה ללא DB בשרת. (לא ספציפית ל react, יכולת כללית מובנית של JS).

localStorage

שמירה offline של key/value בתוך הדפדפן לטווח ארוך (עד שימחק ידנית או התערבות אחרת של היוזר)

שימושים:

1. כל דבר שרוצים שיישמר אחרי שסוגרים את הדפדפן
2. כל דבר שלא רוצים לשתף (כלומר לא לשלוח לשרת)
3. לדוגמא אנשי קשר

סוגי נתונים שאפשר לשמור:

* Strings גם ה value וגם ה key או משהו פשוט להמרה ל string כמו מספר
* מה עם רשימה?

תרגיל משולב react עם Localstorage

נבנה קומפוננטה ששומרת שם וגיל של משתמש.

* שני inputs
* אחרי שהיוזר מכניס את הגיל והשם הם נשמרים ב localstorage