25.02.2024

שיעור 9

**useContext**

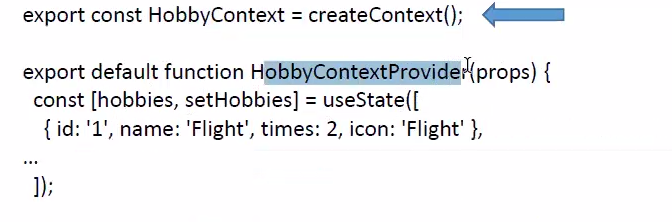
useContext - האופציה לשים סטייט באופן גלובלי.   
למנוע הרבה מאוד פרופס וuplifting.

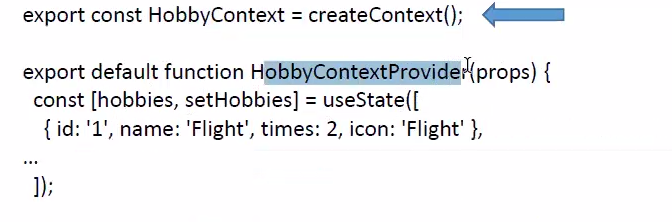
Redux – שיטת עבודה "design pattern", mobX.   
בהתאם לזה יש לריאקט את **Context**:

יוז קונטקסט מוציא סטייט החוצה כך שהוא יהיה מחובר לכל קומפוננטה.

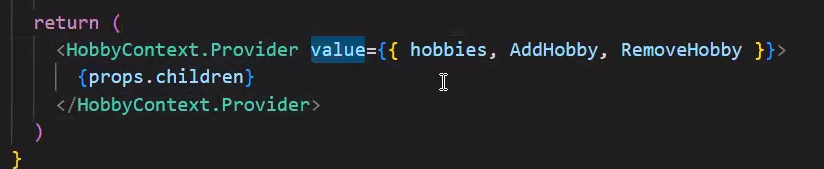
בעצם כל מי ש"מאזין" לסטייט הזה ידע להשתנות.

\*\*לא לשים כל דבר בקונטקסט. אם יש תקשורת בין 2-3 קומפוננטות אז נשתמש בuplifting, כי זה משויך רק אישי לקומפוננטות. קונטקסט הוא משהו גלובלי גדול יותר, שנשתמש רק ברגע שיש שימוש עם כמה קומפוננטות שרחוקות אחת מהשנייה מבחינת תקשורת.

מייצרים קומפוננטה שהיא תהיה רק לקונטקסט, רק לסטייט הגלובלי.   
  
שלבים:  
1. כותבים פונקציה שתביא את הקונטקסט createContext() – צריך לייבא אותה מריאקט. קונטרול ספייס

  
2. מייצרים קומפוננטה שנקראת  
hobbyContextProvidor  
  
פה יישב הסטייט הגלובלי  
(שכתוב עם const מתחת).

3. בקומפוננטה הזאת של contextProvidor שמים בreturn (לפי מה שרשמנו בשלב 1) – היה את הhobbyContext שמקושר לפונקציית create.  
- עושים נקודה Providor.  
רושמים value (מילה שמורה) – שמביאה לי את היכולות הבאות: ומפרטים את היכולות (הפונקציות שאפשר יהיה להשתמש)

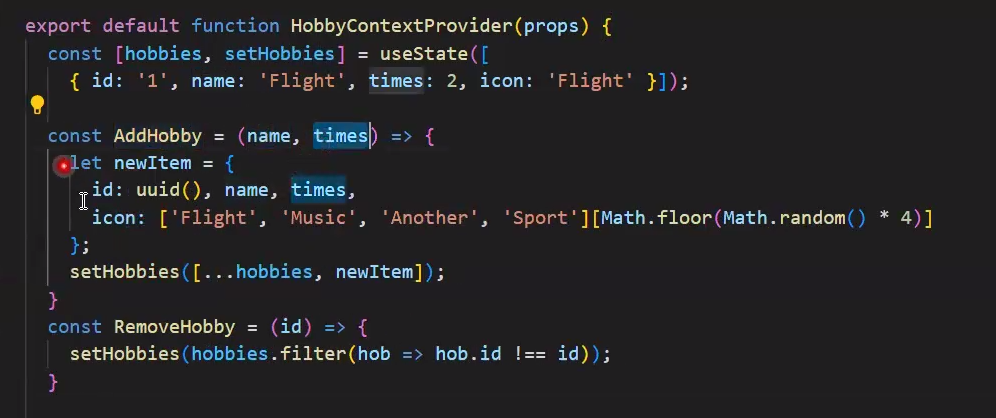


מי שירצה יוכל לראות את רשימת הhobbies (בא מהסטייט שהיה בסעיף 2), להוסיף hobby ולהסיר.

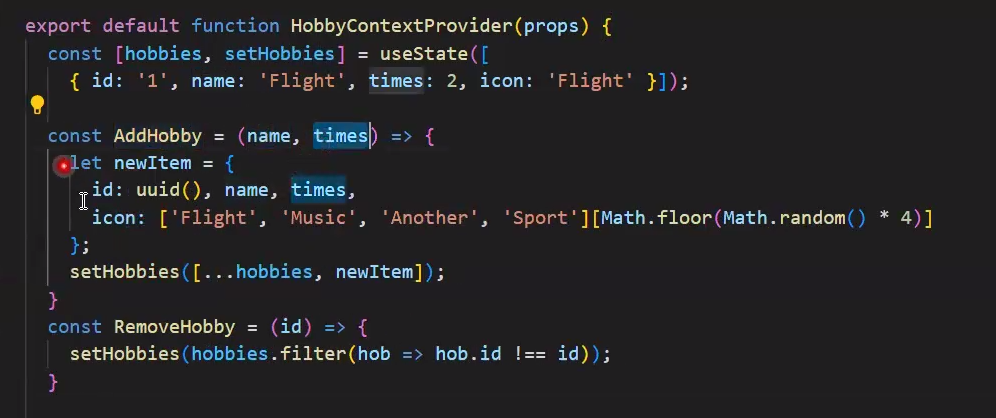
זה עוטף את הchildren – כלומר – כל מי שנעטף על ידי זה, יוכל לקבל את היכולות האלו.

A white background with black text

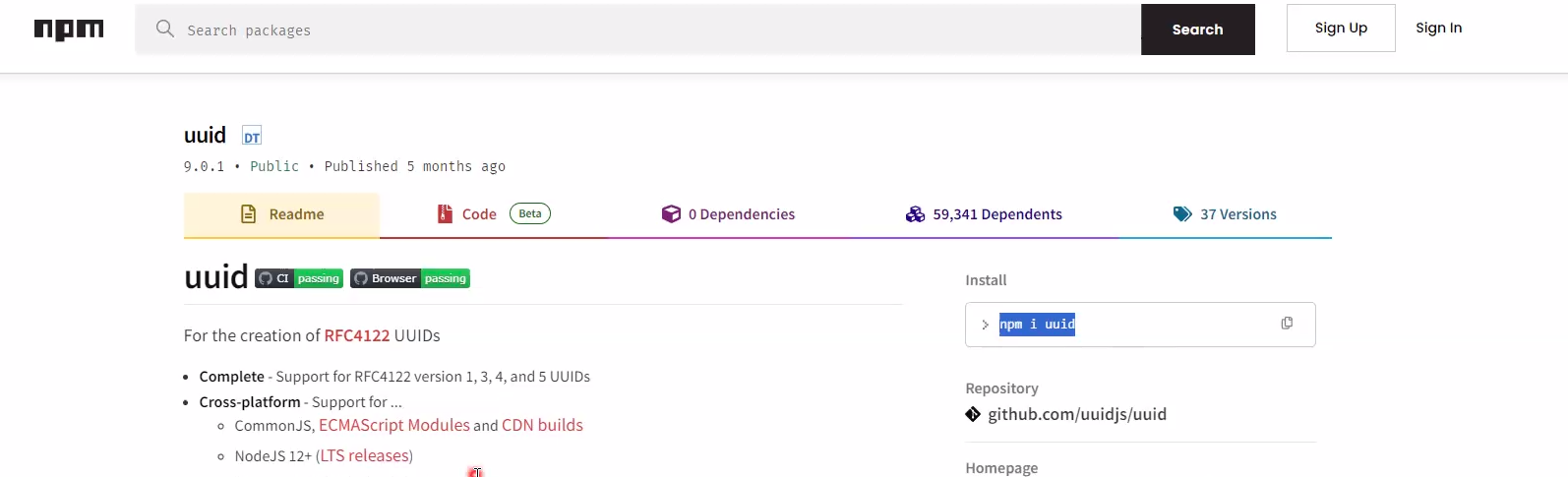
Description automatically generated

4. יוצרים את היכולות/פונקציות שאמרנו בreturn: שמים אותן מתחת לשורה של יצירת הסטייט  


צריך לשלוח לו props  
value וגם props.children – זה מילים של השפה.  
בid של הnewItem הוא מקבל uuid()



**כדי להתקין ספרייה של ID ייחודי:**

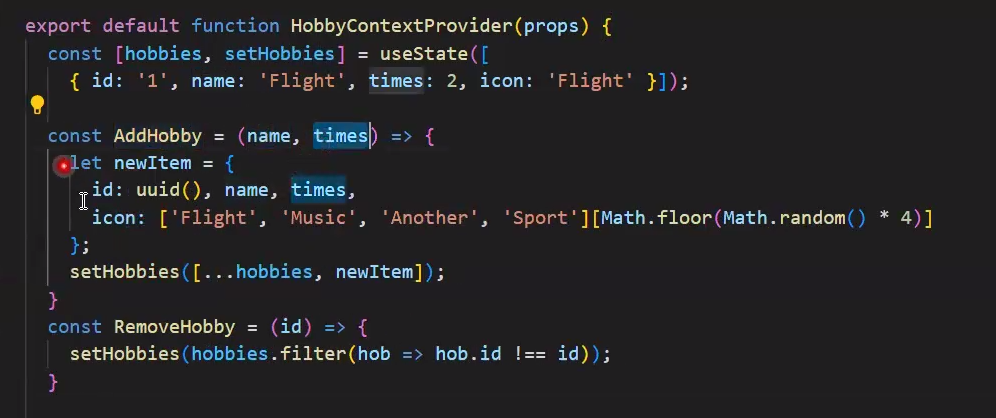
****הולכים לאתר הזה.  
מתקינים npm I uuid

\*לא מתקינים ספריות ברגע שאני עם השרת פתוח לכן תמיד עושים קונטרול C

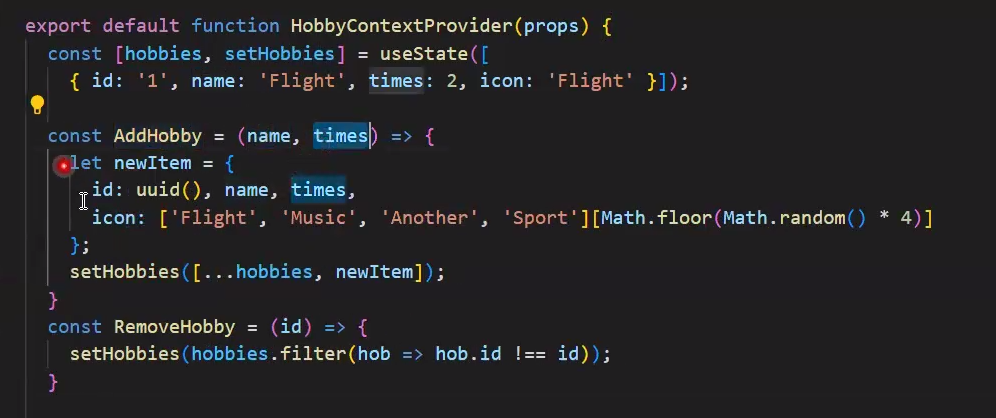
import uuidfrom 'uuid';

npm I uuid

נותן לי ID ייחודי – זה מביא משהו ארוך כזה מכוער עם אותיות ומספרים ו"-"

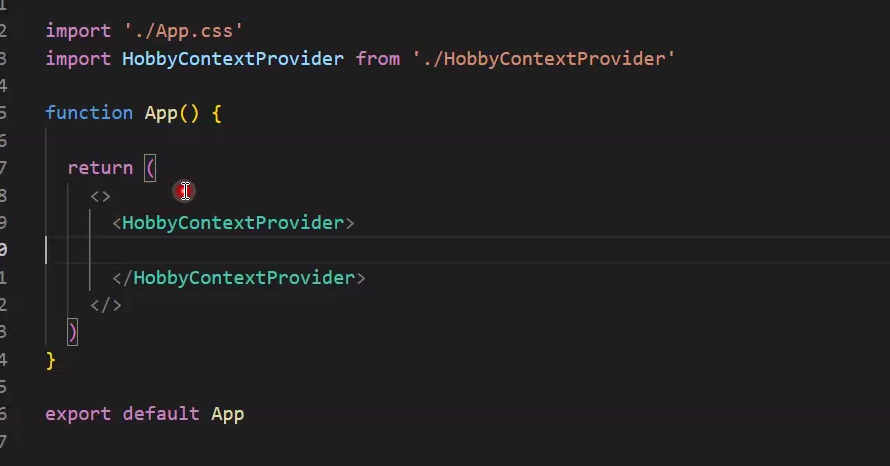


השורה של math random מביאה מספר בין 0 ל3. ובוחר אייקון רנדומלית מהרשימה. Math random מביא מספר בין 0 ל1. אח"כ מכפילים ב4, ועושים floor זה נותן בין 0 ל3.



מוסיף לרשימת הhobbies את הhobby החדש.

אפשר לייצר כמה קונטקסטים. אם אני רוצה שכולם ייגשו לזה, שמים את זה בAPP:

שמים בתוך APP עטיפה פנימית בתוך הfragment <> </> בתוך זה שמים <HobbyContextProvider> מעל ומתחת לקוד.  


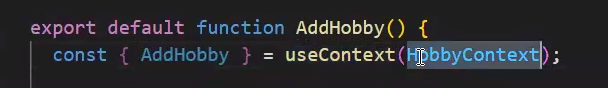
במצב הזה כל מי שבתוך הירוק, יוכל לגשת לhobbies לקונטקסט שיצרתי. בפנים שמה את רשימת הקומפוננטות.

ההבדל בין אספורט לדיפולט:

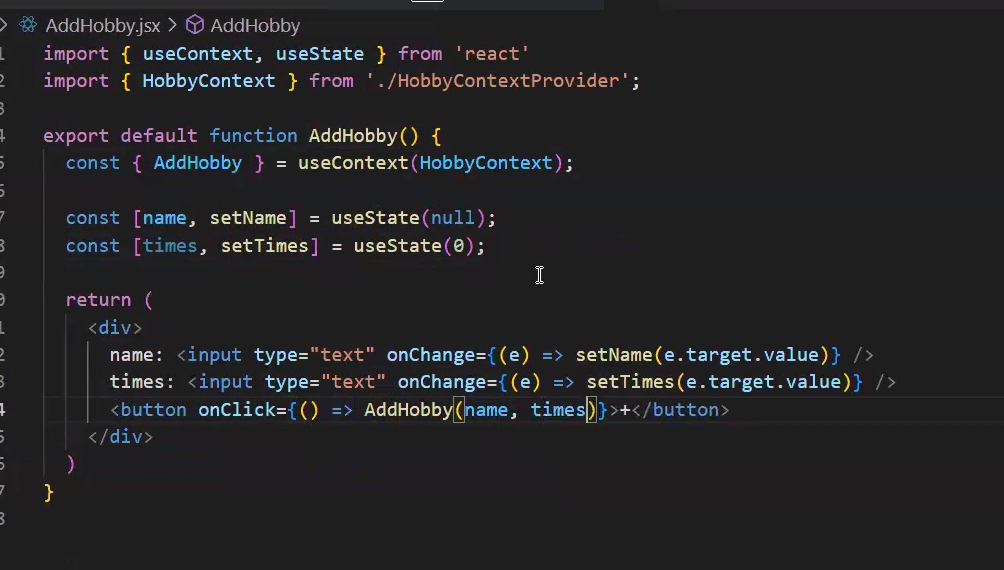
אקספורט – אם יש לי כמה אקספורטים אני יכולה לכתוב רק על אחד דיפולט. כשזה בלי סוגריים אז זה ייקח את הדיפולט. ואם אני רוצה משהו שהוא לא דיפולט אז אני אכתוב {} סוגריים מסולסלים ובתוכו את השם.

**כדי להביא את הקונטקסט:**

1. כותבים useContext ואת השם של הקונטקסט ורושמים שאנחנו רוצים את הפונקציה addHobby.  
   ובתוך הסוגריים של useContext רושמים את הקונטקסט שממנו רוצים להביא את זה (hobbyContext)



ואז אפשר להשתמש בפונקציה בOnClick למטה:



ולשלוח לו את name, times מהפונקציה של הקונטקסט (סעיף 4 למעלה).