Bütün todo’ları bir diziye aktaracağız. Bunun için ;

 (bu daha sonra değişecek.)

Tanımlıyoruz. Bu dizinin içindeki objelerin içeriği bu örnek için ;

İd,

isDone(tıklanıp tıklanmadığını gösterir boolean)

Text.value olacak.

Herbir diziyi obje olarak localStorage’a atmak için ;

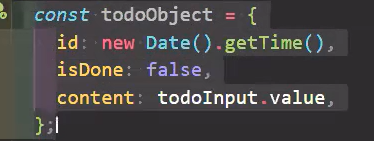
Obje oluştur.

Diziye sakla

Diziyi localStorage’a at

Bunun için tıkladığın yerin altına (btn.addeventlistener)

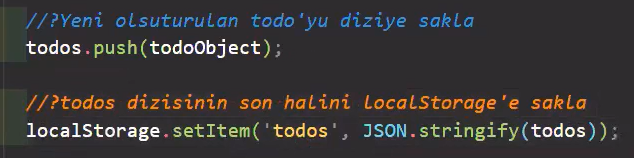
Bir obje oluştur.



Burada id’ye unique bir değer vermek için date metodu kullanıyoruz. isDone değeri default olarak false oluyor. Onu daha sonra değiştireceğiz. Content ise inputun içeriği olacak.

Sonra bunu todos dizisine aktarmamız lazım.

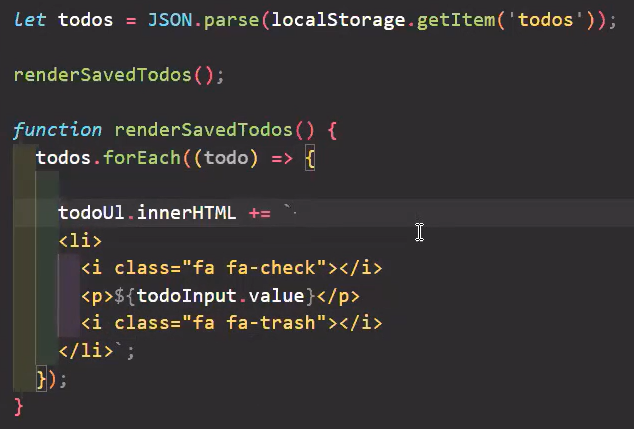
Sonra bunu localStorage’e aktarmamız gerekiyor.



Burada todoObject’i todos’a push’luyoruz ve string formatına çevrilen veriyi localStorage’a atıyoruz.

Cari durumda girilen listeleri localStorage’a atma işlemi bu kadar. Ama Programı kapatıp açtığımızda verileri otomatik çekip liste olarak göstermesini sağlamak için bir fonksiyon yazıyoruz.

Tanımlamaların hemen altına (let todos = [] altına) fonksiyonu yazıyoruz. Bunu let todos = ‘in içine yazmaya karar verdik. (ama bunu da değiştireceğiz. Çünkü ilk başta localStorage boşsa null döner. Bundan dolayı veri yoksa boş array tanımla dememiz gerekir. Aşağıda açıklaması olacak. \*1 )

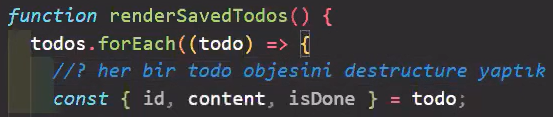


Burada localStorage’dan tüm verileri alacağız. getItem ile aldık. Ama Dom’a yazmamız lazım. Bunun için fonksiyon yazdık. Fonksiyonun içine forEach ile her bir objeyi dolaş diyoruz. Herbir objeyi dolaşıp içindeki bilgileri bize istediğimiz formata çevirecek.

todoUl kısmını aşağıdan kopyalayıp yapıştırmıştık. ${todoInput.value} kısmı bizim inputa girdiğimiz değer olduğu için onu değiştirmemiz gerekir. Onun yerine todo.content yazacağız. 

Çünkü o veri localStorage’ta content içerisinde yazıyor. 

Burada Destructuring kullanacağız.. bu yöntem daha kolay. forEach fonksiyonunun hemen altına ;

,

Yazacağız. Ama destructuring yaparsak artık todo.content yazmamıza da gerek kalmaz. Sadece content yazmamız yeterlidir.

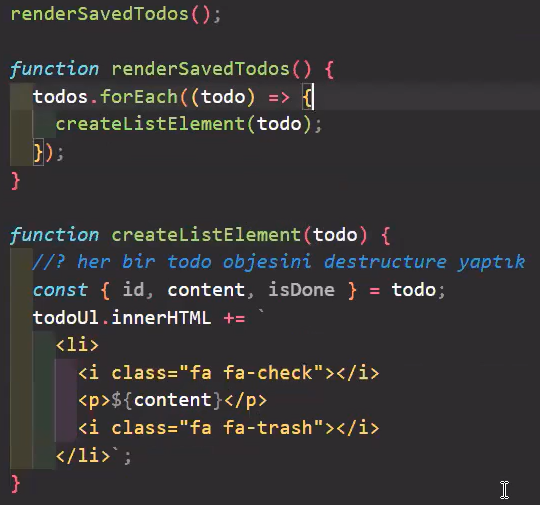
\*1 : Yukarıdaki açıklamanın devamı olarak ;



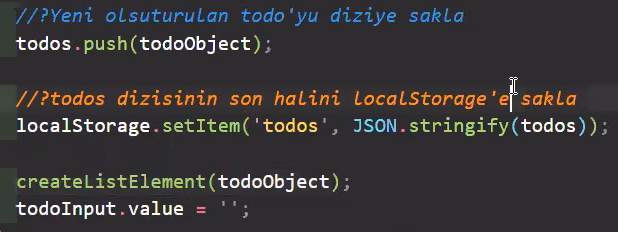
yazarak, eğer localStorage’da bilgi varsa onu getir. Eğer bilgi yoksa boş array oluştur. diyoruz.

(Kullanılan yöntem short-circuit)

Bu aşamada bitti.

Burada sarı olan kısmı tekraren yazmamak için fonksiyon içerisine aldık.

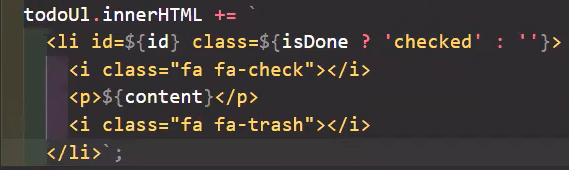
Bunu aşağıdaki sarı olan DOM kısmına da uygulayabiliriz. Orada argüman kısmına todoObject yazmamız gerekir.



Aşağının son hali de bu şekilde.

**isDone** ile devam edeceğiz. isDone’ı sisteme entegre etmemiz gerekir. .checked şeklinde bir class verecek.

True ise checked olacak. False ise bir şey vermeyecek.



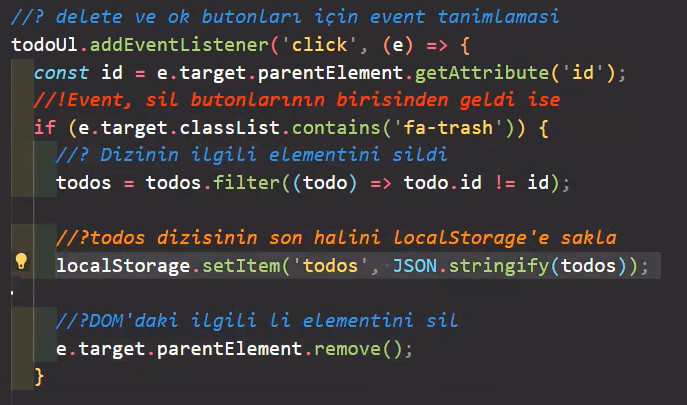
Burada yine short-circuit ile isDone True ise class’ına checked ekle. Yoksa bir şey ekleme dedik. Bununla beraber id ekledik. (localStorage kısmını aşağıda yapacağız.)

**Silme isteği geldiğinde;**

Todos listesinden ilgili kısmı silmemiz lazım.

localStorage’a bunu atmamız lazım.

Aynı zamanda bu değişikliği DOM’a uygulamamız lazım.



Bu kısımda id tanımlıyoruz. (id’yi getAttribute’u id olanı bana getir diye çağırıyoruz).

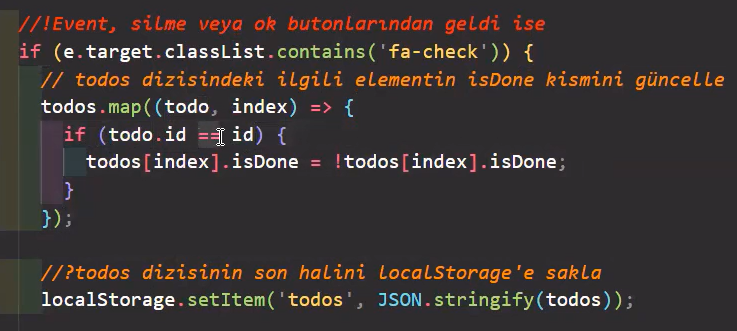
Sonra if bloğu ile “bana id’si click işlemi yaptığım ile aynı olmayan tüm li’leri döndür diyoruz. Böylece id’si eşit olan, yani tıkladığımız silinmiş oluyor. Bu standart bir yazımdır.

Sonra local storage’dan siliyoruz. Sonra DOM’dan siliyoruz.

Şimdi isDone kısmını güncelleme yapacağız ;

Map metodu ile todo içerisinde dolaşıp aradığımız id’li dizinin isDone’ını değiştir diyeceğiz.

Bunun yeri 



Burada map ile todo objesine girdik. İd’si eşleşen objenin indexi isDone olanını -**değili-** ile değiştirdik..

Yani manuel bir toggle yapmış olduk.

**localStorage bitti.**

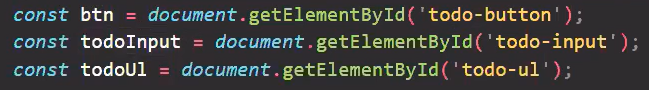
**Aşağıda ek olarak app.js’yi nasıl hazırladığımıza ait not var.**

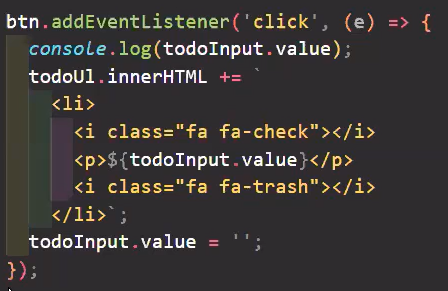
**EK**

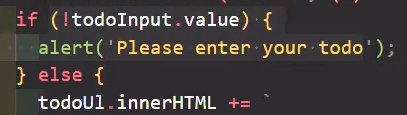
**App.js’yi nasıl hazırladık?**

Değişkenleri tanımladık.

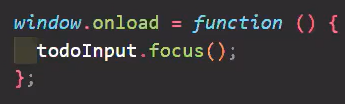
Bunun içerisinde buton, input ve todo kısmı var.





Sonra butona tıkladığımızda şunları yapsın diye addEventListener yapıyoruz. Ve kolay yol olarak innerHTML tanımladık. Ve akabinde inputu sıfırlasın diye boş string attık. Burada ek olarak bir şey yazılmamışsa uyarı versin if bloğu yazdık. 

Aynı işlemi klavyeden de yapsın diye “inputa klavyede enter’a tıklandığında” kodunu yazdık.

Birde tanımlamaların altına sayfa yüklendiğinde inputa focuslansın diye bu kodu yazdık.

Bundan sonra todoUl’ye capturing metodu kullanarak sil ve checked işlemlerini yaptıracağız. Ul’ye addeventlistener atıyoruz. Sonra if ile istediğimiz işlemleri yaptırıyoruz.

