**23 OCAK 2023 DERSİ (JS-19): checkout page Project**

**25 OCAK 2023 ilave ders**

****

LMS Link == > <https://lms.clarusway.com/mod/page/view.php?id=39080>(23 Ocak)

LMS Link == > <https://lms.clarusway.com/mod/page/view.php?id=39212> (25 Ocak)

Instructors -- Ryan Daniel + Steve Mentor

ÖZET:

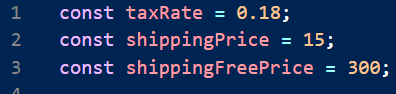
Bir e-ticaret sitesinin alışveriş sepeti görünümüne sahip bir sayfa tasarlıyoruz. Üst kısımda ürün resmi, birim fiyat, miktar artma ve azaltma butonları, alt kısımda üst bölümden çekilen bilgilerle sepet fiyat hesaplamaları yazdırılacaktır. Bazı değişkenleri localStorage altında saklayacağız.

HTML yapısında stillendirme için kolaylık olması açısından fazladan DIV konteynere kullanılmış- bu daha az olabilirdi –

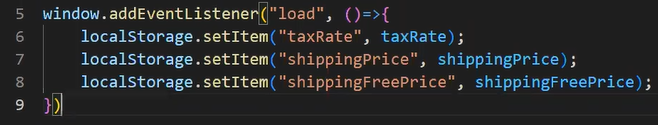
EKSİ ve ARTI butonları “fontawesome” dan temin edildiği için CDNJS’den fontawesome linkini dosyamıza bağlıyoruz.

CSS dosyasında ilave satırlar mevcut, dosyanın sahibi, bu sayfanın öncesi “register form” biçimlendirmesi içinde aynı dosyaya yazmış ve “media query” yazarak ilaveler eklemiş. Aslında ihtiyacımız olan dosya 100 satırı geçmez.

--- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- --- --- --- --- ---



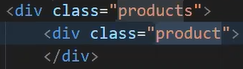
* GLOBAL değişken olarak vergi oranını, kargo ücretini ve ücretsiz kargo limitini tanımlıyoruz. Gerçek uygulamalarda bunların diğer sayfalardan da ulaşılabilir olması gerekiyor. Bunu simüle etmek için bunları data base’den alıyormuş gibi localStorage’dan (getItem’la) çağıracağız.



* Önce bu değişkenleri sayfa yüklendiğinde (load) localStorage’a setItem’la gönderiyoruz. SET’de key:value şeklinde veri ikilileri kullanılır. Burada ilk değer key’lerin adları yukarıdaki DEĞİŞKEN adlarıyla aynı olmak ZORUNDADIR. Value isimlerini ise başka adla tanımlayabiliriz.

Derste JS hata vermemesi için değişken tanımlama ve bunları addEventListener’la localStorage’a gönderen satırları hoca yoruma aldı. Program tamamlandığında bu satırların tekrar açılması gerekir, aksi halde alt taraftaki ücret hesaplayan bölüm eksik çalışır.

--- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- --- --- --- --- ---

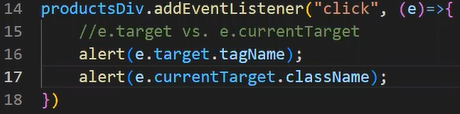


* Ürünlerle ilgili artma/azaltma işlemlerini yapıldığı (TIKLAMA işlemi) en yakın kapsayıcı statik alanı seçeriz. Burada “.products” DİV, yeni bir ürün eklenirse ve çıksa da fonksiyon etkilenmesin ve bu alanda görevine devam etsin.  JS’de bu kapsayıcıyı “productsDiv” adını verdiğimiz bir değişkene aktardık. Burada “CAPTURING” – yukarıdan aşağıya/dıştan içe” yapıyoruz. Bunun tersi “BUBLING” ise içeriden dışarıya doğrudur.



bu değişkene “click” eventListener tanımladık.

--- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- --- --- --- --- ---

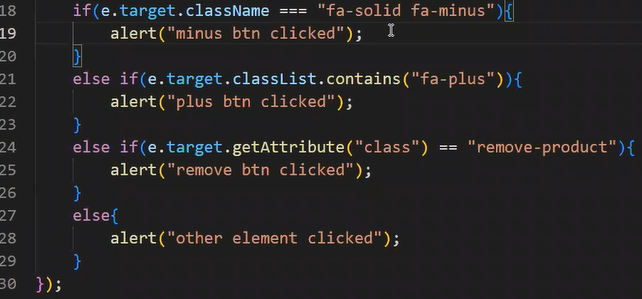


* target ve currentTarget arasındaki fark,

TARGET == > tıklanan elemanın kendisini verir

CURRENTTARGET == > tıklanan elemanı değil, addEventListener tanımlanmış kapsayıcıyı verir (burada “.products Div)

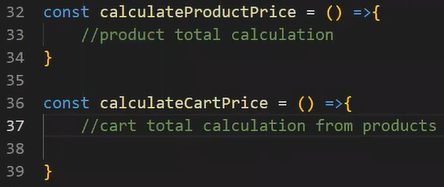
--- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- --- --- --- --- ---



* Yakalanacak bütün “click” ler için bir çatı yapı oluşturduk. if (eksi butonu) // else if (artı butonu) // else if (remove butonu) // else (bunlar harici)

ÖRNEK olması için bunların her birisini farklı yöntemlerle yakaladık. “className” kullanılırken bire bir aynısı yazılmalıdır. “classList.contains” ise class’lardan en az bir tanesini alması yeterli, “getAttribute” sadece “key” çağırabilir, burada class adlı key’in “remove-product” eşit olanını seçiyor.

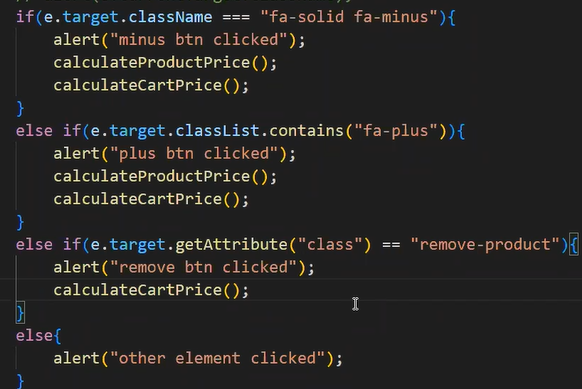
--- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- --- --- --- --- ---



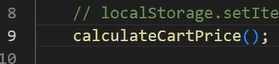
Bu aşamada

* 1. Ürün fiyatını hesaplayan (calculateProductPrice)
* 2. Sepet fiyatı hesaplayan (calculateCartPrice) iki adet fonksiyona ihtiyaç olduğu görülüyor. GLOBAL’de bu 2 fonksiyon TASLAK olarak yazdık.

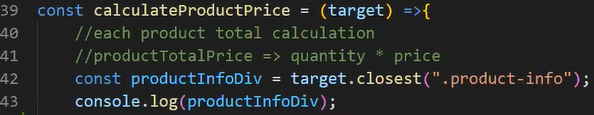
--- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- --- --- --- --- ---



* Hangi butona bastığımızda hangi fonksiyonu çağıracağımıza karar veriyoruz. Arttırma ve azaltma işlemlerinde hem ürün hem sepet fiyatı, SİLME işleminde ise sadece sepet fiyatını hesaplamak en mantıklısıdır.

 window load aşamasında ekranda sayılı ürünler olacağı için buna bağlı sepet hesaplaması çıkması için bu fonksiyonu window.load’da çağırıyoruz. Bunu yapmazsak butonlardan birine basana kadar sepet hesaplama göstermez.

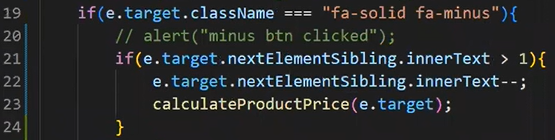
--- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- --- --- --- --- ---



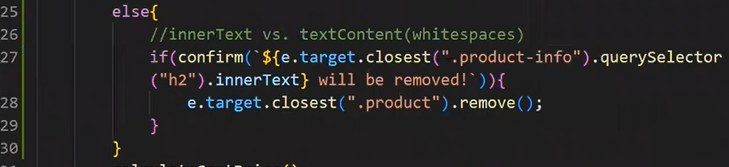
*  Ürün fiyatı hesaplama fonksiyonuna “event.target” property veriyoruz ve böylelikle bu fonksiyon sadece “click” yapılan ürün için hesaplama yapacak. Ayrıca ürün miktarı ve ürün birim fiyatı alıp bunları çarparak hesap yapabilmek için bu bilgileri kapsayan en yakın statik kapsayıcı olan “.product-info” için bir değişken (productInfoDiv) tanımladık, hesaplama bu haliyle aşağıda devam edecek.

--- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- --- --- --- --- ---

* EKSİ butonunu düzenliyoruz. Burada tekrar “if” “else” yapısı kullanacağız. Eğer eksi butonunun sonrasındaki (nextElementSibling) elemanın değeri 1’den büyükse bunu bir kez azaltıyoruz ve ürün fiyat hesaplama fonksiyonunu (calculateProductPrice(e.target)) burada çağırıyoruz.



* ELSE şartında (bu değer 1’den büyük değilse) tekrar “if” ile bu ürünü silinecek anlamında DOĞRULAMA (CONFIRM) gönderiyoruz. Kullanıcı silmeyi onaylarsa ürün sayfadan siliniyor. Confirm ifadesinin kullanıcı tarafından daha anlaşılır olması için ürünün h2 ile yazılmış adını seçiyoruz. Bunu seçmek için tıklanan (e.target) yere en yakın parent (closest) .product-info DIV’ine çıktık ve onun altındaki h2 elementini seçtik. Doğrudan h2 seçemeyiz çünkü h2 EKSİ butonuyla parent/child ilişkisinde değildir.

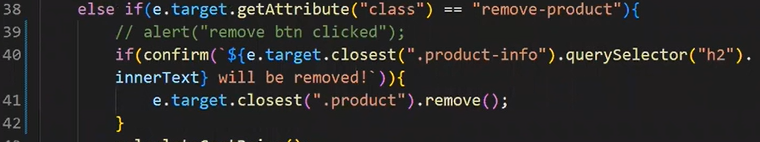


Burada “innerText” kullanıldı, “textContent” tercih edilmedi. Textcontent eğer varsa BOŞLUKLARI da (whitespaces) kapsar.

CONFIRM ve ALERT ikazlarının göze daha hitap eden çeşitleri için sweetalert2 kütüphanesi kullanılabilir.

Link == > <https://sweetalert2.github.io/>

--- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- --- --- --- --- ---



Yukarıda kullanılan DOĞRULAMA -CONFIRM kod bloğunu SİLME butonuna olduğu haliyle kopyaladık.

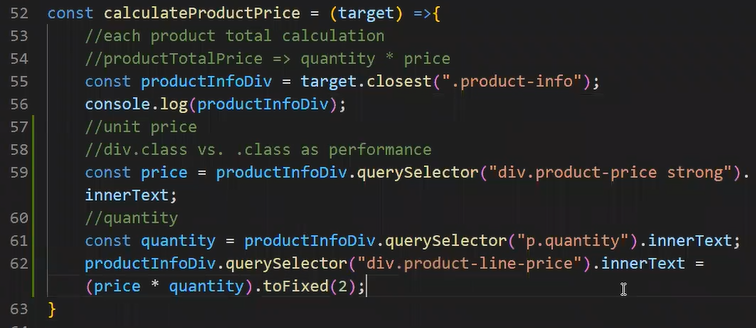
--- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- --- --- --- --- ---

* ARTI butonunu düzenliyoruz. Tıklanan artı butonun parent elementine git ve onun “.quantity” class sahip child elementinin innerText’ini arttır. Ürün fiyat hesaplama (calculateProductPrice(e.target)) ve sepet fiyat hesaplama (calculateCartPrice()) fonksiyonlarını çağır.

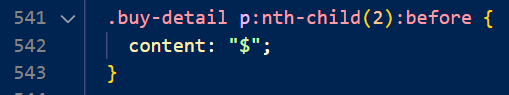
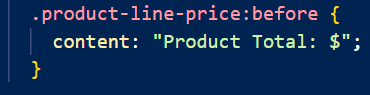


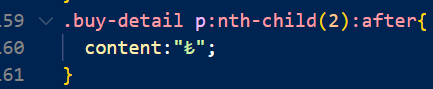
--- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- --- --- --- --- ---

* Ürün fiyat hesaplama (calculateProductPrice(e.target)) fonksiyonuna son halini veriyoruz. “price” adında bir değişkene ürün fiyatının HTML yapısındaki adresini yakalayıp innerText ile aldık. “quantity” adlı başka bir değişkene ürün miktarının HTML adresini yakalayıp innerText ile adedini aldık. Bu iki değişkenin çarpma işlemiyle hesaplanan ürün fiyat bilgisinin yazılacağı “product-line-price” DIV’ine bunu yazdırdık. Sayının virgül (,) den sonraki kısmını 2 hane ile sınırladık. (toFixed(2))



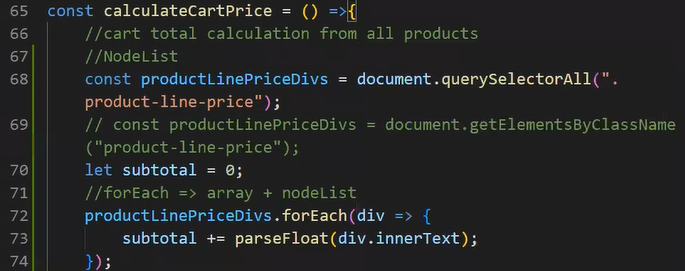
* HTML’de fiyat bilginin yanına ($) (₺) gibi para birimi işaretleri koyarsanız innerText bunu yakaladığında “number” gibi işlem yapamaz ve hesaplamalarınız “NaN” olur. Bundan en kolay kaçınma yöntemi CSS dosyasında ilgili yere :before veya :after ile içerik eklemektir.



--- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- --- --- --- --- ---

* SEPET fiyatını hesaplıyoruz. “querySelectorAll” operatörü ile bütün ürün ara TOPLAM larını yakalayıp (productLinePriceDivs) bir değişkene aktardık. ALL operatörü bize dönüş olarak bir NODELIST (array-like // dizimsi) oluşturmuştu. Bunda forEach() metodu kullanabiliriz. Önce ara toplamı (subtotal) SIFIR olarak belirliyoruz ve bunları “number” laştırıp (parseFloat) topluyoruz. Bunu yapmazsak JS burada “string concat” yapar.



Bu işlemin REDUCE metodu ile yapılması (Steve Mentor dersinden- seçilen değişken ismi farklıdır)



--- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- --- --- --- --- ---

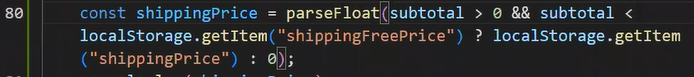
* VERGİ lerin hesaplanması, daha önce localStorage’a gönderdiğimiz “taxRate” verisini GET ile çekiyoruz. Basit çarpma işlemi ile buluyoruz. GET içine yazılacak parametre SET içine yazılan KEY ve aynı zamanda değişkenin adıdır.



--- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- --- --- --- --- ---

* KARGO bedeli hesaplama, eğer toplam fiyat 0’dan büyük ve localSorage’da tanımlı limit (shippingFreePrice) fiyattan (burada 300) düşük ise localStorage’da kayıtlı KARGO bedeli (shippingPrice) EĞER değilse 0’dır. “parseFloat”ile bunu sayı olarak hesaplamasını sağladık.

TERNARY YAPISIYLA YAZILDI.



--- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- --- --- --- --- ---

* TOPLAM hesaplıyoruz. Basit toplama işlemi



--- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- ---- --- --- --- --- --- --- --- --- ---

* Sepet için bulduğumuz verileri HTML’e yazdırıyoruz. Her bir satır için ayrı bir seçme yöntemi kullanıldı.



ürün toplamı // querySelector seçicisi



Vergi // getElemenyById seçicis



Kargo bedeli // CSS yapısından seçim tekniği olarak querySelector ile ikinci child’ı seç



Toplam ücret // querySelector ile SON child’ı seç

toFixed(2) ile vigülden sonra 2 hane ile sınırlandır.

