JavaScript Becerilerini JSFiddle Kullanarak Geliştirme

Tahmini Süre: 60 dakika

Giriş:

Bu laboratuvar, JavaScript programlama becerilerinizi geliştirmek için pratik alıştırmalar sunmak amacıyla tasarlanmıştır. Odak noktası, gerçek dünya mantıksal problemlerini çözmekte, sizi etkili kod yazmaya ve bunları JS Fiddle'da çalıştırarak çıktıyı görmeye teşvik etmektedir. Bu laboratuvarın sonunda, çeşitli senaryolar için çözümler uygulama yeteneğinizi geliştirmiş olacak ve kodlama becerilerinizde güven kazanacaksınız.

Amaç:

- JavaScript kullanarak problem çözme becerilerini geliştirmek
- Mantıksal programlar yazma ve hata ayıklama pratiği yapmak
- Döngüler, fonksiyonlar ve koşullu mantık kullanarak gerçek dünya çözümlerini nasıl uygulayacağınızı anlamak
- JSFiddle gibi platformlarda kodlama uygulamalarını güçlendirmek

---

Alıştırma 1: Toplam satış tutarını hesaplayın

Problem:
Bir çevrimiçi mağazada çalışıyorsunuz. Göreviniz, belirli bir satış işlemleri kümesi için toplam satış tutarını hesaplayan bir JavaScript kodu yazmaktır.

Girdi ayrıntıları:

- Satış işlemlerini temsil eden bir nesne dizisi. Her nesne aşağıdaki özelliklere sahiptir:
  - `item`: Ürünün adı (string)
  - `quantity`: Satılan birim sayısı (integer)
  - `price`: Birim fiyatı (float)

Çıktı ayrıntıları:

- Toplam satış tutarını temsil eden tek bir sayı

Uygulama adımları:

1. En az 3 örnek nesne içeren bir satış işlemleri dizisi tanımlayın
2. Bu diziyi girdi olarak alan `calculateTotalSales` adlı bir fonksiyon yazın
3. Diziyi döngü ile geçerek toplam satış tutarını hesaplayın
4. Toplam satış tutarını konsola yazdırın

▼ İpuçlarını görmek için buraya tıklayın

- `sales` dizisi, her biri `item`, `quantity` ve `price` özelliklerine sahip olan satış işlemlerini temsil eden nesneleri içerir.
- `sales` dizisindeki her nesneyi geçmek için bir döngü kullanın.
- Her nesne için, `quantity` ile `price`'ı çarpın ve o ürün için toplamı elde edin.
- Toplamları bir değişkende biriktirerek genel satış tutarını elde edin.
- Biriktirilmiş toplamı döndürün ve `console.log` kullanarak görüntüleyin.

▼ Çözüm kodunu görmek için buraya tıklayın

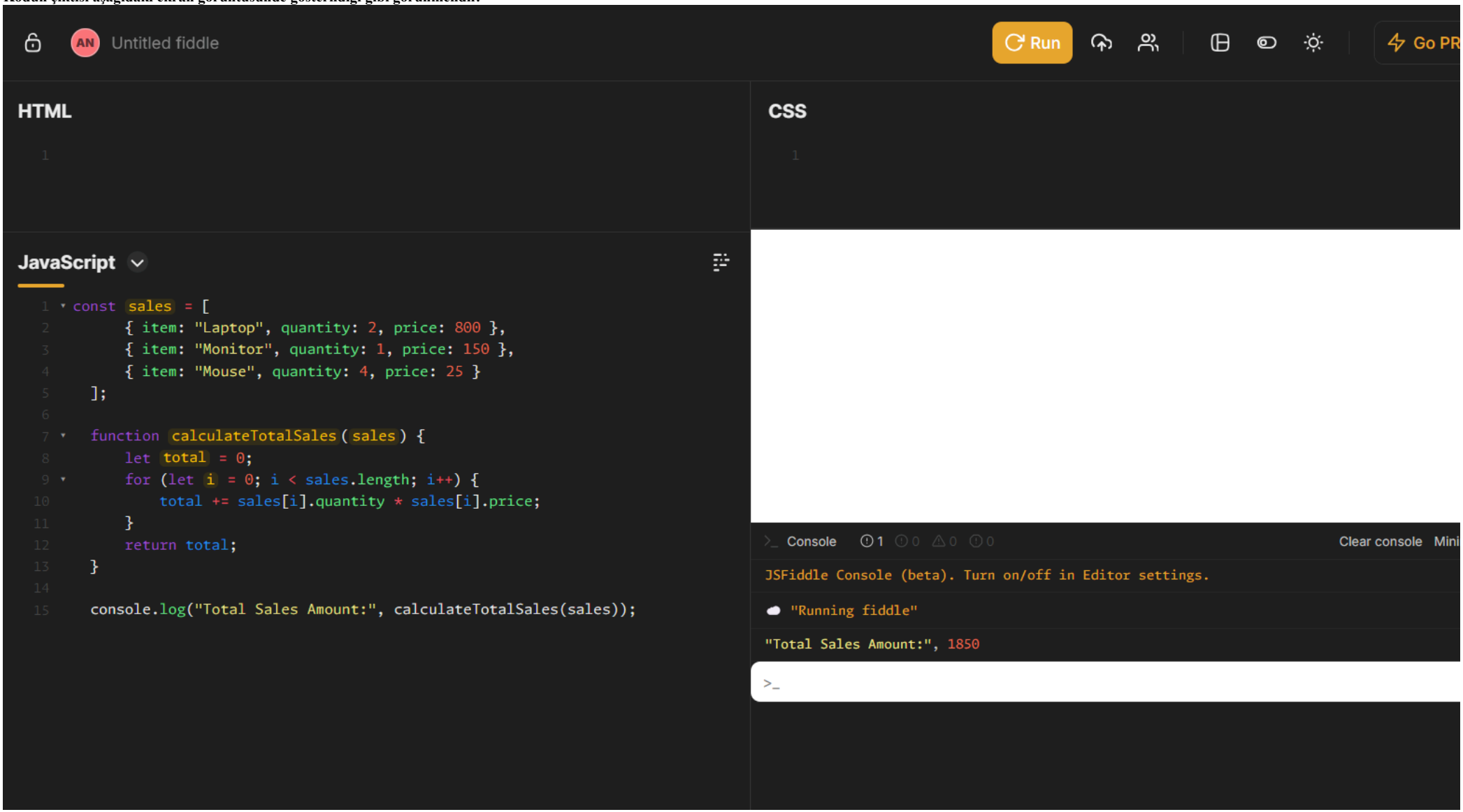
```
const sales = [
    { item: "Laptop", quantity: 2, price: 800 },
    { item: "Monitor", quantity: 1, price: 150 },
    { item: "Mouse", quantity: 4, price: 25 }
];
function calculateTotalSales(sales) {
    let total = 0;
    for (let i = 0; i < sales.length; i++) {
        total += sales[i].quantity * sales[i].price;
    }
```

```
    return total;
}
console.log("Toplam Satış Tutarı:", calculateTotalSales(sales));
```

Programı JSFiddle üzerinde yazın:

- JSFiddle adresine gidin
- Kodu JavaScript bölümüne yazın
- Programı çalıştırmak için `Run` butonuna tıklayın ve sonuçları konsol bölümünde kontrol edin

Kodun çıktısı aşağıdaki ekran görüntüsünde gösterildiği gibi görünmelidir.



Alıştırma 2: Bir sipariş fişi oluşturun

Problem:
Bir müşterinin siparişi için bir fiş oluşturan bir JavaScript programı yazın. Fiş, her bir ürünün adı, miktarı, fiyatı ve toplam maliyetini içermelidir.

Girdi detayları:

- Sipariş edilen ürünleri temsil eden bir nesne dizisi. Her nesne şunları içerir:
  - `item`: Ürünün adı (string)

    - `quantity`: Sipariş edilen miktar (integer)
    - `price`: Birim fiyatı (float)

Çıktı detayları:

- Her ürünün detaylarını ve toplam tutarı gösteren ayrıntılı bir fiş

Uygulama adımları:

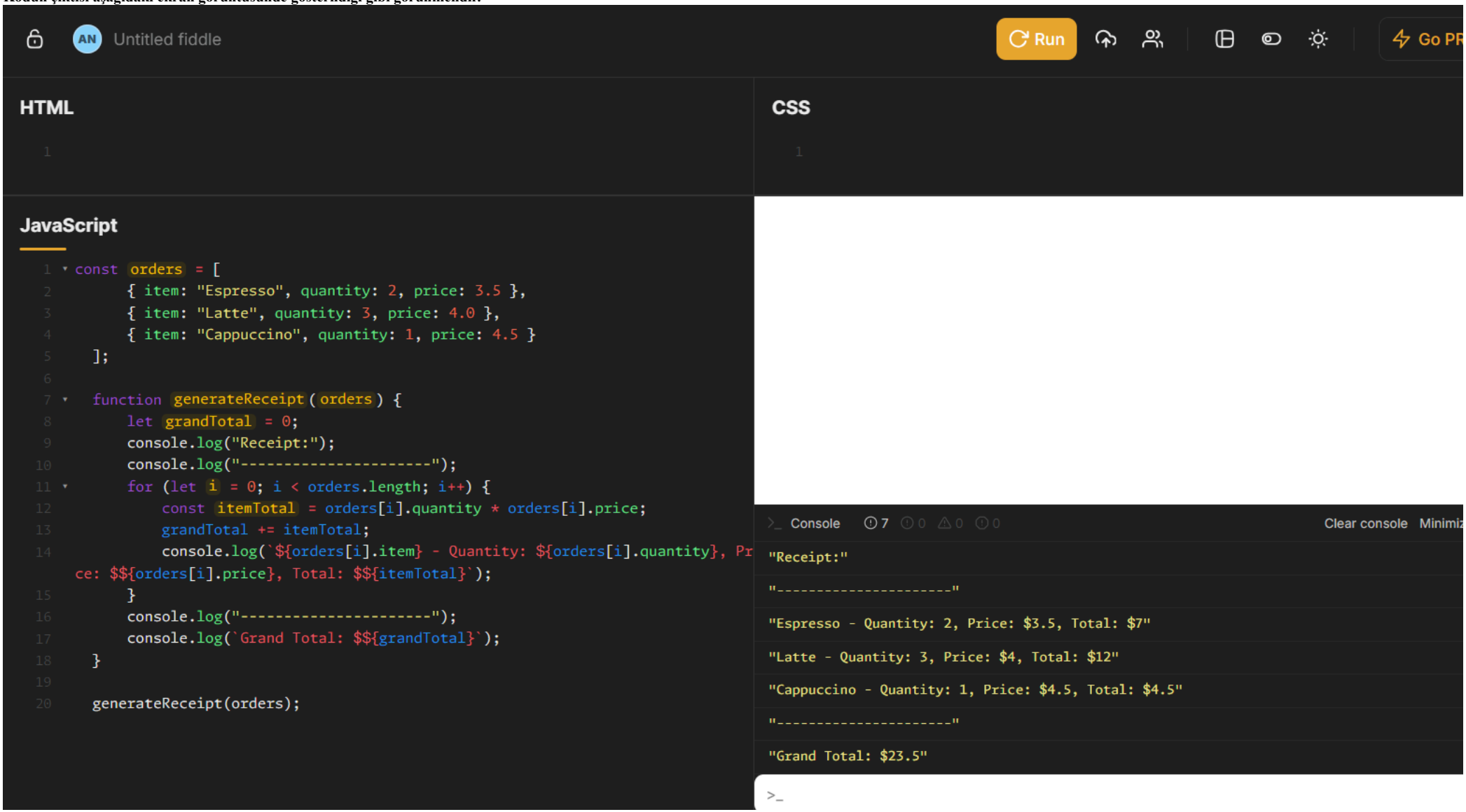
1. En az 3 örnek girişi olan sipariş edilen ürünlerin bir dizisini tanımlayın
2. Bu diziyi girdi olarak alan bir `generateReceipt` fonksiyonu yazın
3. Ürünleri döngü ile geçerek her ürünün toplamını ve genel toplamı hesaplayın
4. Fişi formatlı bir dize olarak yazdırın

▶ İpuçlarını görmek için buraya tıklayın
▶ Çözüm kodunu görmek için buraya tıklayın

Programı JSFiddle'da yazın:

- JSFiddle adresine gidin
- Kodu JavaScript bölümüne yazın
- Programı çalıştırmak için `Run` butonuna tıklayın ve sonuçları konsol bölümünde kontrol edin

Kodun çıktısı aşağıdaki ekran görüntüsünde gösterildiği gibi görünmelidir.



Alıştırma 3: Şifreleri Doğrula

Problem:
Bir liste şifreyi doğrulamak için bir JavaScript programı yazın. Bir şifre geçerli ise:

- Sadece alfanümerik karakterler (harfler ve rakamlar) içermelidir
- En az 8 karakter uzunluğunda, en fazla 20 karakter uzunluğunda olmalıdır

Girdi detayları:

- Şifrelerin (string) bir dizisi

Çıktı detayları:

- Her bir şifrenin geçerli veya geçersiz olduğunu belirten bir mesaj

Uygulama adımları:

1. Örnek şifrelerden oluşan bir dizi tanımlayın
2. Bu diziyi girdi olarak alan `validatePasswords` adlı bir fonksiyon yazın
3. Şifreleri döngü ile geçerek her birini doğrulama kriterlerine karşı kontrol edin
4. Her bir şifrenin geçerli veya geçersiz olduğunu kaydedin

▼ İpuçlarını görmek için buraya tıklayın

- `passwords` dizisi, verilen kurallara göre doğrulanması gereken stringler içerir.
- Geçerli şifreler için deseni tanımlamak üzere bir düzenli ifade kullanın. Desen yalnızca alfanümerik karakterlere izin vermeli ve 8 ile 20 karakter arasında olmalıdır.
- Her bir şifrenin düzenli ifadeyle eşleşip eşleşmediğini kontrol etmek için `test` yöntemini kullanın.
- `passwords` dizisi üzerinde döngü yapın ve her bir şifreyi regex kontrolünün sonucuna göre geçerli veya geçersiz olarak kaydedin.

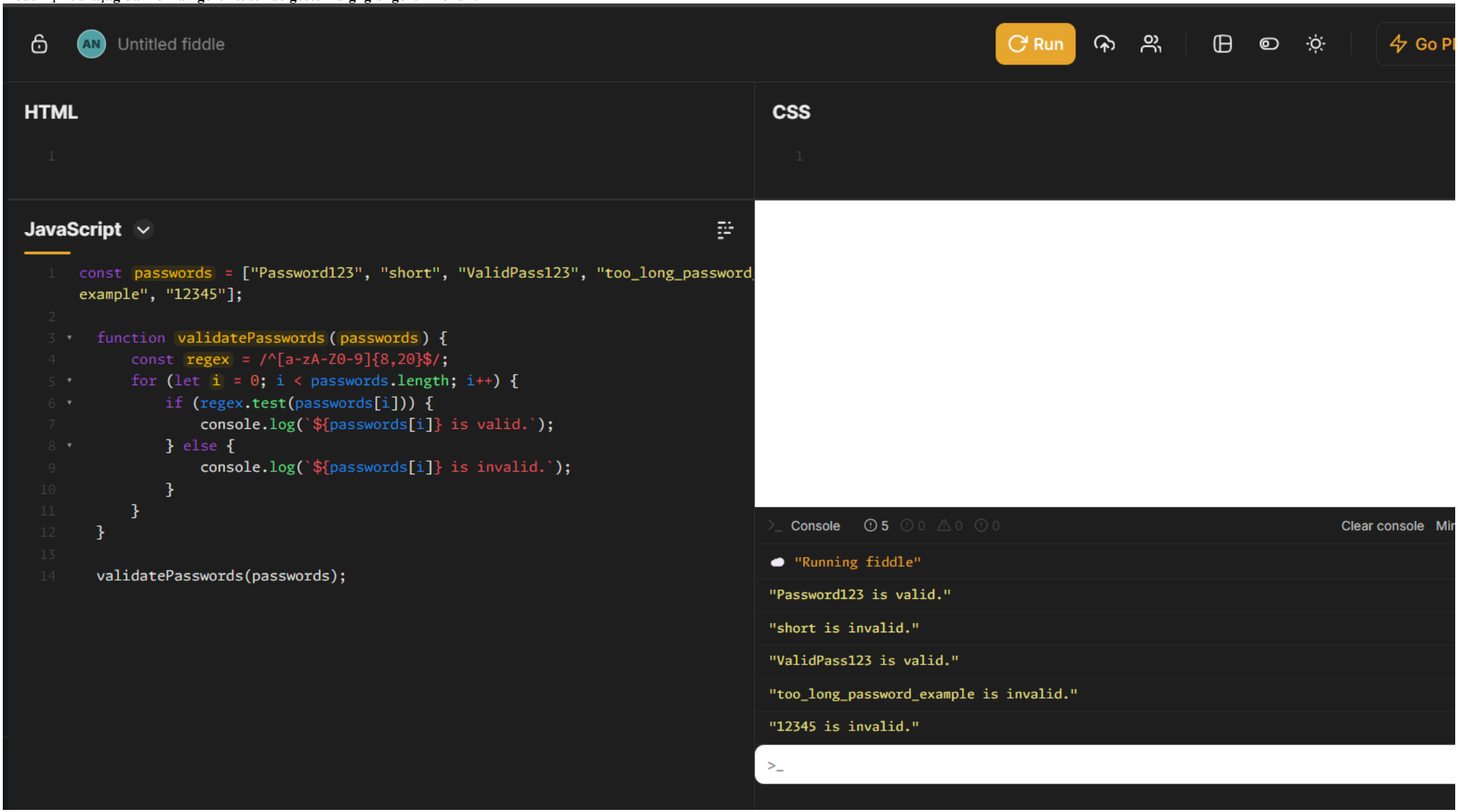
▼ Çözüm kodunu görmek için buraya tıklayın

```
const passwords = ["Password123", "short", "ValidPass123", "too_long_password_example", "12345"];
function validatePasswords(passwords) {
    const regex = /^[a-zA-Z0-9]{8,20}$/;
    for (let i = 0; i < passwords.length; i++) {
        if (regex.test(passwords[i])) {
            console.log(`${passwords[i]} geçerli.`);
        } else {
            console.log(`${passwords[i]} geçersiz.`);
        }
    }
}
validatePasswords(passwords);
```

Programı JSFiddle'da yazın:

- JSFiddle adresine gidin
- JavaScript bölümüne kodu yazın
- Programı çalıştırmak için `Run` butonuna tıklayın ve sonuçları konsol bölümünde kontrol edin

Kodun çıktısı aşağıdaki ekran görüntüsünde gösterildiği gibi görünmelidir.



Alıştırma 4: Ürün stok seviyelerini takip et

Problem:
Bir çevrimiçi perakende şirketinde çalışıyorsunuz. Göreviniz, envanterdeki çeşitli ürünlerin stok seviyelerini takip eden bir JavaScript programı yazmaktır. Program, bir ürünün stokta olup olmadığını kontrol etmeli ve uygun bir mesaj kaydetmelidir.

Girdi detayları:

- Ürünleri temsil eden bir nesne dizisi. Her nesne şunları içerir:
  - `product`: Ürünün adı (string)
  - `stock`: Stokta mevcut birim sayısı (integer)

Çıktı detayları:

- Her bir ürün için ürünün stokta olup olmadığını belirten bir mesaj.

Uygulama adımları:

1. En az 3 örnek ürün ile ürün nesnelerinden oluşan bir dizi tanımlayın
2. Bu diziyi girdi olarak alan `checkStockLevels` adında bir fonksiyon yazın
3. Diziyi döngü ile dolaşarak her ürünün stok seviyesini kontrol edin
4. Ürünün “Stokta” mı yoksa “Stokta Yok” mu olduğunu belirten bir mesaj yazdırın

▼ İpuçlarını görmek için buraya tıklayın

- `products` dizisi `product` ve `stock` özelliklerine sahip nesneleri içerir.
- `products` dizisini dolaşmak ve her ürünün stok seviyesini kontrol etmek için bir döngü kullanın.
- Stok 0’dan büyükse, “Stokta” yazdırın; aksi takdirde “Stokta Yok” yazdırın.

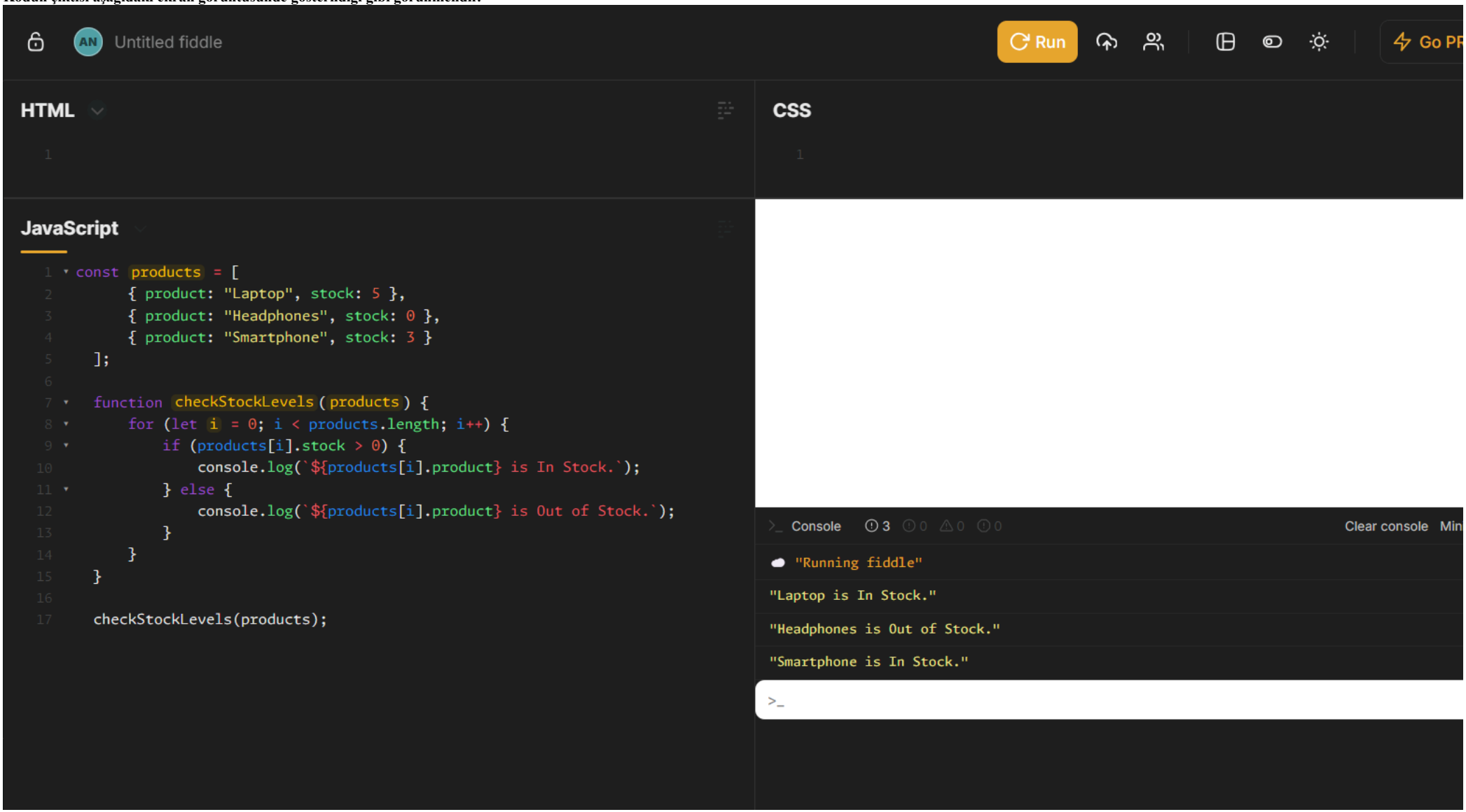
▼ Çözüm kodunu görmek için buraya tıklayın

```
const products = [
    { product: "Laptop", stock: 5 },
    { product: "Headphones", stock: 0 },
    { product: "Smartphone", stock: 3 }
];
function checkStockLevels(products) {
    for (let i = 0; i < products.length; i++) {
        if (products[i].stock > 0) {
            console.log(`${products[i].product} Stokta.`);
        } else {
            console.log(`${products[i].product} Stokta Yok.`);
        }
    }
}
checkStockLevels(products);
```

Programı JSFiddle’da yazın:

- JSFiddle adresine gidin
- Kodu JavaScript bölümüne yazın
- Programı çalıştırmak için `Run` butonuna tıklayın ve sonuçları konsol bölümünde kontrol edin

Kodun çıktısı aşağıdaki ekran görüntüsünde gösterildiği gibi görünmelidir.



Sonuç:

Bu alıştırmalar sayesinde, JavaScript kullanarak orta seviyedeki problemleri çözme pratiği yaptınız. Her görev, doğrulamadan string manipülasyonuna kadar mantıksal düşünmenin farklı yönlerine odaklandı. Programlama becerilerinizi ve güveninizi artırmak için benzer zorluklarla pratik yapmaya devam edin.

Yazar

[Rajashree Patil]

Değişiklik Günlüğü

Tarih	Versiyon	Değiştiren	Değişiklik Açıklaması
2024-27-12	———	Prashant Juyal	QA düzenlemeleri
2024-13-12	1.0	Rajashree Patil	İlk versiyon oluşturuldu