

React'te Bileşenler ve Props: Temel Bilgiler

1. Bileşenler (Components): Temel Yapılar

React'te bileşenler, kullanıcı arayüzünüzü küçük, yönetilebilir parçalara ayırmanın temel yoludur. Bileşenler, genellikle kullanıcı arayüzü elemanlarını tanımlar ve uygulamanızın bir bölümünü temsil eder. Bu bileşenler, genellikle JavaScript fonksiyonları veya sınıfları olarak tanımlanır ve JSX (JavaScript XML) kullanılarak yapılandırılır. JSX, HTML benzeri bir sözdizimi sağlar ve JavaScript kodu içinde HTML öğelerini tanımlamanıza olanak tanır. Bileşenler iki ana türde olabilir: Fonksiyonel Bileşenler ve Sınıf Bileşenleri. Fonksiyonel bileşenler, JavaScript fonksiyonları olarak tanımlanır ve genellikle daha basit bileşenler için kullanılır. Sınıf bileşenleri ise daha karmaşık durum yönetimi ve yaşam döngüsü metodları gerektiren durumlar için kullanılır.

2. Props: Bileşenlere Veri Geçişi

Props, bileşenlere veri geçişi yapmak için kullanılan bir özelliktir. Props, 'properties' kelimesinin kısaltmasıdır ve bileşenlere dışarıdan veri sağlamak için kullanılır. Bir bileşene veri gönderirken, bu verileri props olarak tanımlarız. Props, bileşenin nasıl görüneceğini veya nasıl davranacağını belirlemek için kullanılır ve bileşenin dışarıdan aldığı veri olarak düşünülebilir. Props kullanarak, bileşenler arasında veri paylaşımı yapılabilir ve aynı bileşeni farklı veri setleri ile kullanabilirsiniz. Props'lar, bileşenlerin yeniden kullanılabilirliğini artırır ve kodunuzu daha modüler hale getirir. Props'lar sadece veri sağlamakla kalmaz, aynı zamanda bileşenlerin işlevselliğini kontrol etmenize olanak tanır.

3. Bileşen İçinde Props Kullanımı

Bileşenler içinde props kullanarak bileşenlerin dinamik ve yeniden kullanılabilir olmasını sağlarız. Props, bileşenin içine veri geçmek ve bu verileri bileşenin görünümünde veya davranışında kullanmak için kullanılır. Böylece, aynı bileşeni farklı veri setleri ile kullanarak, farklı durumlar için özelleştirilmiş kullanıcı arayüzleri oluşturabilirsiniz. Props kullanımı, bileşenler arasında veri paylaşımını kolaylaştırır ve uygulamanın yönetimini daha verimli hale getirir. Props ile bileşenler arasında veri geçişi yaparken, veri kaynağı bileşenden ayrılmış olur ve bileşenler sadece ihtiyaç duyduğu veriyi alır. Bu yaklaşım, bileşenlerin bağımsız ve tekrar kullanılabilir olmasını sağlar.

Basit Bir Kullanıcı Kartı Bileşeni Örneği

Bu örnekte, kullanıcı adı, yaş ve fotoğraf gibi bilgileri içeren bir kullanıcı kartı bileşeni oluşturacağız. Bileşen, dışarıdan aldığı props ile kişiselleştirilecek.

```
1 import React from "react";
2
3 function UserCard({ name, age, city }) {
4   return (
5     <div>
6       <h2>{name}</h2>
7       <p>Yaş: {age}</p>
8       <p>Şehir: {city}</p>
9     </div>
10  );
11 }
12
13 function App() {
14   return (
15     <div>
16       <UserCard name="Volkan" age={20} city="İstanbul" />
17     </div>
18   );
19 }
20
21 export default App;
```

VOLKANKIYCAK.COM.TR