Conditional Rendering

if, ternary (? :) ve && operatörü ile koşula göre farklı arayüz göstermek, ve koşula göre farklı component'ler döndürmek -- basit örneklerle.

Başlangıç 9 dk

Son güncelleme: 18 Ağustos 2026
Çevrimiçi oku: https://www.learnforgex.com/tr/topics/conditional-rendering

---

Conditional Rendering

JSX dersinde, süslü parantez `{ }` içine yalnızca "sonuç üreten" bir ifade konabildiğini, bu yüzden `if` yazamayıp bir ternary kullandığımızı kısaca görmüştük. Bu ders, koşula göre farklı arayüz göstermeyi ("conditional rendering") baştan sona, State ile birlikte işliyor.

if ile Conditional Rendering

`if`, JSX'in `{ }` içine doğrudan yazılamaz -- ama `return`'den ÖNCE, normal bir JavaScript değişkenine karar vermek için kullanabilirsin:

```
function IfConditionalExample({ isLoggedIn }) {
  // if does NOT PRODUCE A VALUE, so it can't be written directly inside
  // JSX's { }. Instead, we decide on a plain variable BEFORE the return.
  let message;

  if (isLoggedIn) {
    message = "Welcome!";
  } else {
    message = "Please log in.";
  }

  return <p>{message}</p>;
}
```

Burada `if`, hangi metnin `message` değişkenine atanacağına karar veriyor; JSX'in kendisi hâlâ tek bir `{message}` ifadesi.

Ternary (? :) ile Conditional Rendering

Ternary, bir SONUÇ ÜRETTİĞİ için `{ }` içine doğrudan yazılabilir -- `IfConditionalExample` 'daki birkaç satırı tek satıra indirir:

```
function TernaryConditionalExample({ isLoggedIn }) {
  // The ternary (? :) PRODUCES A VALUE, so it can be written directly
  // inside { }. This condenses the four lines from IfConditionalExample
  // into a single line.
  return <p>{isLoggedIn ? "Welcome!" : "Please log in."}</p>;
}
```

İki seçenek arasında (bu VEYA şu) karar verirken ternary, `if`'ten daha kısa ve JSX içinde daha doğal.

&& Operatörü ile Conditional Rendering

Bazen iki seçenek arasında değil, "ya bir şey göster ya da hiçbir şey gösterme" arasında seçim yaparsın. Bunun için `&&` kullanılır:

```
function AndOperatorConditionalExample({ hasNewMessage }) {
  // The && operator: renders the right side if the left side is "truthy",
  // and renders nothing if it's "falsy" (false, 0, "", null, undefined).
  // Useful when there's no "else" branch like in a ternary -- i.e. for
  // "either show this or show nothing" situations.
  return (
    <div>
      <p>Inbox</p>
      {hasNewMessage && <p>You have a new message!</p>}
    </div>
  );
}
```

`&&`'nin solu "truthy" (gerçek, dolu, sıfır olmayan) ise sağ taraf render edilir; solu "falsy" (`false` , `0` , `""` , `null` , `undefined`) ise React hiçbir şey render etmez. **Yaygın bir hata:** sol taraf bir sayıysa ve o sayı `0` olabiliyorsa, `0 && <p>...</p>` ekranda gerçekten `0` yazısını render eder -- çünkü `0` "falsy" olsa da JSX içinde render edilebilir bir değerdir. Bunu önlemek için sayıyı `hasNewMessage

> 0` gibi bir karşılaştırmayla açıkça boolean'a çevirmek daha güvenli.

Conditional Component'ler

Bazen koşula göre değişen şey tek bir metin değil, tamamen farklı bir component'tir:

```
function LoadingMessage() {
  return <p>Loading...</p>;
}

function WelcomeMessage({ name }) {
  return <p>Welcome, {name}!</p>;
}

function ConditionalComponentExample({ isLoading, name }) {
  // Instead of just a single piece of text, you can return an entirely
  // DIFFERENT component depending on the condition -- both are meaningful
  // components in their own right.
  if (isLoading) {
    return <LoadingMessage />;
  }

  return <WelcomeMessage name={name} />;
}
```

`ConditionalComponentExample`, koşula göre ya `<LoadingMessage />` ya da `<WelcomeMessage />` döndürüyor -- her ikisi de kendi başına anlamlı, bağımsız component'ler. Bu, Component Composition dersinde gördüğümüz "küçük component'lerden büyük arayüzler kurma" fikrinin, koşullarla birleşmiş hâli.

Özet ve Terimler Sözlüğü

Conditional rendering, koşula göre farklı içerik göstermenin yoludur. `if`, JSX dışında (return'den önce) bir değişkene karar vermek için kullanılır; ternary iki seçenek arasında, `&&` ise "göster ya da hiç gösterme" durumunda JSX'in içinde doğrudan kullanılır. Koşula göre tamamen farklı component'ler de döndürülebilir.

Terimler Sözlüğü

Conditional Rendering — Bir koşula göre farklı içerik (metin, element, component) render etme.

Truthy / Falsy — JavaScript'te bir değerin `if` / `&&` gibi bir mantık bağlamında sırasıyla "doğru" ya da "yanlış" gibi davranması; `false`, `0`, `""`, `null`, `undefined` falsy'dir, geri kalan her şey truthy'dir.

`&&` **Operatörü** — Solundaki ifade truthy ise sağındaki ifadeyi döndüren, değilse hiçbir şey render etmeyen JavaScript operatörü.