



레이아웃 사용하기

레이아웃을 사용하면 여러 페이지에서 공통으로 사용되는 UI 요소를 쉽게 관리할 수 있어요. 헤더, 네비게이션 바, 푸터와 같은 공통 컴포넌트를 레이아웃으로 구성하면 코드 중복을 줄이고 일관된 사용자 경험을 제공할 수 있어요.

레이아웃 파일 만들기

레이아웃은 `_layout.tsx` 파일을 생성해서 구현할 수 있어요. 이 파일의 위치에 따라 적용되는 범위가 달라져요.

tsx

```
import { PropsWithChildren } from "react";

export default function Layout({ children }: PropsWithChildren) {
  return <>{children}</>;
}
```

레이아웃 적용 범위

레이아웃은 파일 위치에 따라 다른 범위에 적용돼요.

- `pages/_layout.tsx` : 모든 페이지에 적용
- `pages/about/_layout.tsx` : `intoss://{서비스명}/about` 하위의 모든 페이지에 적용

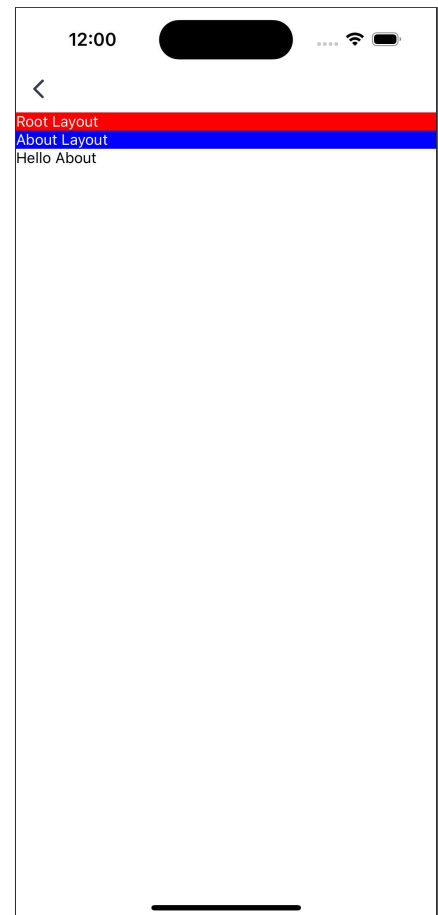
레이아웃은 중첩해서 사용할 수 있어요. 페이지에 여러 레이아웃이 적용될 때는 상위 디렉토리부터 순차적으로 적용돼요.

```
pages/
├─ _layout.tsx      // 전역 레이아웃
├─ about/
```

```

|   └─ _layout.tsx    // about 섹션 레이아웃
|   └─ index.tsx      // about 메인 페이지
|   └─ team.tsx       // 팀 소개 페이지
└─ index.tsx          // 메인 페이지

```



예를 들어, 위와 같은 구조로 `_layout.tsx` 가 구성되어 있을 때, `about/team.tsx` 페이지는 다음 순서로 레이아웃이 적용돼요.

1. `pages/_layout.tsx` (최상위 레이아웃)
2. `pages/about/_layout.tsx` (about 섹션 레이아웃)
3. `pages/about/team.tsx` (실제 페이지 컴포넌트)

이렇게 레이아웃을 중첩해서 사용하면 전역적으로 필요한 UI 요소와 특정 섹션에서만 필요한 UI 요소를 효과적으로 구성할 수 있어요.

레이아웃 예시

전역 레이아웃

모든 페이지에 적용되는 레이아웃을 만들어 보세요.

pages/_layout.tsx components/Header.tsx components/Footer.tsx

tsx

```

import { PropsWithChildren } from "react";
import { View } from "react-native";
import { Header } from "../components/Header";
import { Footer } from "../components/Footer";

export default function Layout({ children }: PropsWithChildren) {
  return (
    <View style={{ flex: 1 }}>
      <Header />
      {children}
      <Footer />
    </View>
  );
}

```

```
);
}
```

섹션별 레이아웃

특정 섹션에만 적용되는 레이아웃을 만들어 보세요.

pages/about/_layout.tsx

components/AboutSidebar.tsx

tsx

```
import { PropsWithChildren } from "react";
import { View } from "react-native";
import { AboutSidebar } from "../../components/AboutSidebar";

export default function AboutLayout({ children }: PropsWithChildren) {
  return (
    <View style={{ flexDirection: "row" }}>
      <AboutSidebar />
      <View style={{ flex: 1 }}>{children}</View>
    </View>
  );
}
```

레이아웃에서 쿼리 파라미터 받아오기

레이아웃에서 쿼리 파라미터를 받아오려면 `useParams` 훅을 사용할 수 있어요. `useParams` 는 현재 화면의 파라미터를 읽어서 객체 형태로 반환해요.

useParams 훅 사용 예시

다음은 `_layout.tsx` 파일에서 `useParams` 훅을 사용하는 예제예요. URL 쿼리 파라미터로 전달된 `title` 값을 기반으로 화면 상단에 제목을 동적으로 표시해요.

pages/_layout.tsx

tsx

```
import { useParams } from "react-native-bedrock";
import { PropsWithChildren } from "react";
import { View, Text } from "react-native";
```

```
export default function Layout({ children }: PropsWithChildren) {
  // 현재 화면의 파라미터를 가져와요.
  const params = useParams({ strict: false });

  // 'title' 파라미터를 가져오고 기본값을 설정해요.
  const title = params?.title ?? "기본 제목";

  return (
    <View style={{ flex: 1 }}>
      { /* 동적으로 생성된 헤더 */ }
      <View style={{ padding: 16, backgroundColor: "#f0f0f0" }}>
        <Text style={{ fontSize: 20, fontWeight: "bold" }}>{title}</Text>
      </View>
      { /* 자식 컴포넌트를 렌더링 */ }
      <View style={{ flex: 1 }}>{children}</View>
    </View>
  );
}
```

레퍼런스

- [화면 이동하기](#)
- [쿼리 파라미터 사용하기](#)

Previous page

[이미지 불러오기에 실패했을 때 처리하기](#)

Next page

[스크롤 뷰에서 요소 감지하기](#)