

useContext

August 17, 2026

useContext

Kind: Function

Source: `src/jsx/context.ts`

Part of: `Jsx`

Read the current value of a context created with `createContext`.

Safe to call from async components after `await`. See `createContext` for the per-runtime isolation guarantees.

`useContext` reads the current value from a context created with `createContext`. It can be called in synchronous or asynchronous components, including after an `await`, while keeping context values isolated per runtime.

Signature

```
function useContext(context: Context<T>): T
```

Parameters

Name	Type
<code>context</code>	<code>Context<T></code>

Returns: `T`

Diagram

```
graph LR
  A[createContext] --> B[Context]
  B --> C[Context.Provider]
  C --> D[Component]
```

```
D --> E[useContext]
E --> F[Current context value]
```

Usage

```
import { createContext, useContext } from 'hono/jsx'

const ThemeContext = createContext('light')

const ThemeLabel = async () => {
  await Promise.resolve()

  const theme = useContext(ThemeContext)

  return <span>Theme: {theme}</span>
}

const App = () => (
  <ThemeContext.Provider value="dark">
    <ThemeLabel />
  </ThemeContext.Provider>
)
```

AI Coding Instructions

- Create contexts with `createContext` before reading them with `useContext` .
- Pass the context object returned by `createContext` directly to `useContext` .
- Read context inside a component that is rendered below the matching `Context.Provider` .
- `useContext` may run after `await` in an async component.

Used by

7 references from 7 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (7)

- `Props` — `src/jsx/base.ts :27`
- `childrenToString` — `src/jsx/components.ts :14`
- `FormContext` — `src/jsx/dom/hooks/index.ts :24`
- `clearCache` — `src/jsx/dom/intrinsic-element/components.ts :18`

- `HasRenderToDom` — `src/jsx/dom/render.ts` :30
- `title` — `src/jsx/intrinsic-element/components.ts` :121
- `StreamingContext` — `src/jsx/streaming.ts` :30