wawiwa
React
2022 © Wawiwa Tech | Confidential

wawiwa
5 - DOM virtual
2022 © Wawiwa Tech | Confidential

wawiwa

Introducción

El DOM virtual es un concepto fundamental de React.
En este capítulo aprenderemos:

- Recordaremos que es DOM
- Qué es el DOM virtual
- ¿Cómo funciona el DOM virtual?
- Pros y Contras del DOM Virtual

wawiwa

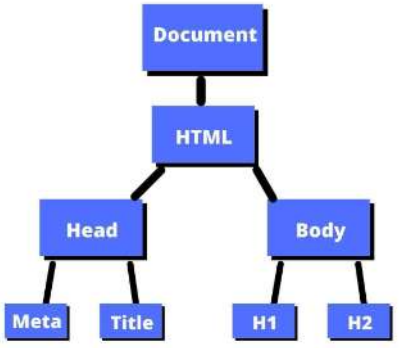
¿Qué es el DOM?

Para entender el DOM virtual y aprender por qué React lo implementa, necesitamos entender que es el DOM del navegador.

Cuando una página web es solicitada por un usuario, el navegador usualmente recibe del servidor un documento HTML para esa página. Entonces el navegador The browser then **construye una estructura lógica como un árbol** del documento de HTML para que el usuario pueda ver la página solicitada en el lado del cliente.

DOM significa Document Object Model (Modelo de objetos de documento).

DOM sirve como una **interfaz para el documento web** para que Javascript y otros lenguajes puedan acceder e **interactuar** programáticamente y **manipular** los contenidos del documento. Por ejemplo, los programadores pueden usar los APIs del DOM para agregar o borrar elementos, borrar sus apariencias y ejecutar las acciones del usuario con los elementos de la web.



wawiwa

¿Cómo funciona el DOM?

Los marcos de JavaScript generalmente actualizar todo el DOM enseguida, causando que el app sea lenta.

¿Cómo funciona el DOM virtual?

Cada vez que el DOM virtual es actualizado, React **compara** el DOM virtual con una instancia del estado inicial antes de actualizarlo. Usando esta comparación, React js automáticamente averigua cual parte de tu componente de React necesita ser actualizado.

Interesantemente, React usa a lo que se le llama **diffing algorithm (algoritmo de diferenciación)** para hacer esto. Entonces una vez que React diferencia, actualiza los componentes que deben ser actualizados con sus nodos actualizados.

wawiwa

DOM Virtual

Como su nombre lo implica, el DOM virtual es una representación "virtual" del DOM actual.

El término “Virtual" se refiere a una **réplica** considerablemente más **liviana** del DOM actual. Esto crea una forma de objetos que pueden ser guardados en la memoria del navegador.

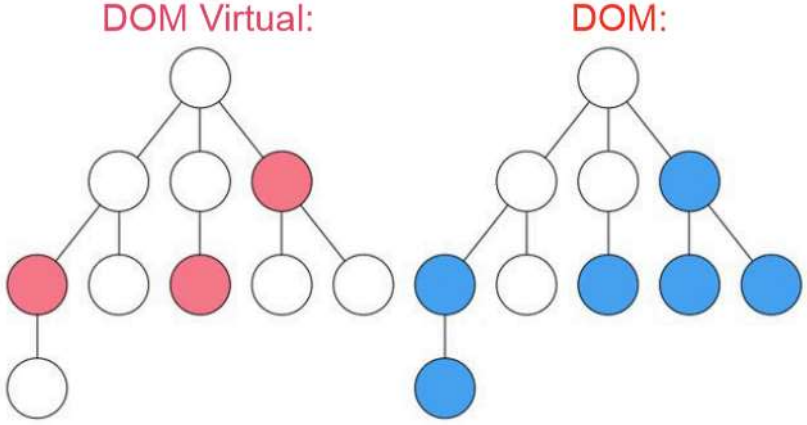
Una idea comúnmente errónea es que el DOM virtual es más rápido o que compite con el DOM real - esto está mal. De hecho, las operaciones del DOM virtual soportan o complementan las operaciones del DOM real. Provee un mecanismo que permite que el DOM actual **calcula la mínima operación del DOM cuando procesa la interfaz del usuario**.

wawiwa

DOM Virtual

Este enfoque nos permite especificar en qué estado queremos que esté la interfaz del usuario, y React después lo implementa. Abstrae las manipulaciones DOM manuales del desarrollador y nos ayuda a escribir código más predecible y tranquilo para que podamos centrarnos en la construcción de componentes.

Tampoco tenemos que preocuparnos de las transiciones del estado. Cuando actualizamos el estado, React se asegura que el DOM coincida con el estado. El siguiente ejemplo presenta la diferencia de renderizado:



wawiwa

Pros y contras del DOM Virtual

Ventajas:

- Velocidad y desempeño aumentado
- Menos pesado
- Simple y claro
- Increíble algoritmo de diferenciación
- Puede ser usado no para React, también para otros entornos de desarrollo

Desventajas:

- Problemas importantes de uso de memoria, ya que los diferentes algoritmos necesitan comparar elementos constantemente para saber qué componentes deben actualizarse o cambiarse.
- No es fácil de integrar con otros entornas de desarrollo
- No puedes usarlo ni manejar motores de plantillas.

Incluso con las desventajas, el DOM virtual siempre es la primera opción para aumentar el desempeño y la velocidad.

wawiwa

Reglas para escribir JSX

- Componentes y tags HTML deben siempre estar cerradas< />
- Algunos atributos como **“class”** y **“className”** deben ser escritos con camelCase
- No podemos retornar más de un elemento HTML a la vez, así que asegúrese de incluirlos dentro de una etiqueta principal:

Como alternativa, puedes envolverlos en tags vacíos:

```
return (
  <div>
    <p>Hello</p>
    <p>World</p>
  </div>
);
```

```
return (
  <>
    <p>Hello</p>
    <p>World</p>
  </>
);
```

```
return (
    <p>Hello</p>
    <p>World</p>
);
```