Curso de React.js con TypeScript

@jonalvarezz 

Tipado en React

Jonathan Alvarez
@jonalvarezz

TypeScript gana la batalla contra Flow.

Ya no hay duda que se volvió el lenguaje tipado para JavaScript por defecto, y en gran parte gracias a Angular y el gran apoyo de la comunidad.

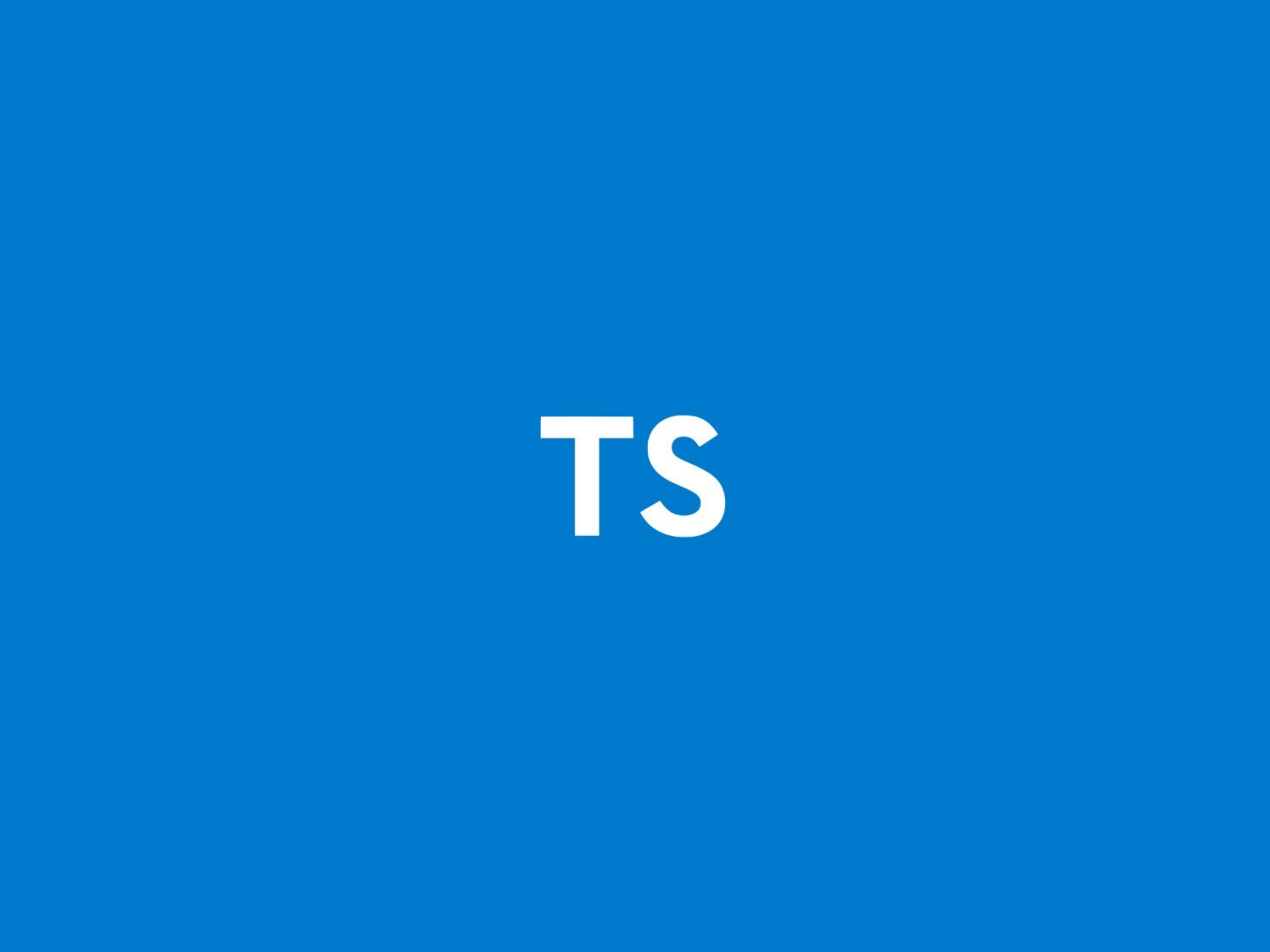
Si tuviera una app en Flow, migrar a TypeScript sería un plan a realizar pronto.

Translate Tweet

3:10 PM · Dec 31, 2019 · Twitter for iPhone

El presente del Frontend es TypeScript


NEXT
.JS


TS


tailwindcss

Tipado implícito vs. Tipado Explícito

Explícito vs. Implícito

- **Explícito**
  Lo **ves** en el código.

- **Implícito**
  Lo **sabes** luego de ver o leer el código.

```jsonc
// tsconfig.json
{
  // ...
  "compilerOptions": {
    "strict": true,
    "noImplicitAny": true,
    "noImplicitReturns": true,
    // ...
  }
}
```

Creando una app con React y TypeScript


npx create-next-app@latest --ts --use-npm

Diferentes formas de definir un componente

El objeto props y children

State con Tipos primitivos

State con otros Tipos

React y el DOM

Tipos para Eventos y Callbacks de Escuchadores

Tipos para Referencias

Componentes que extienden elementos del DOM

Componentes que extienden elementos DOM – Parte 2

Diseño de API para componentes

- Sigue la API web nativa

```
// ❌
<Image imageURL="/assets/image.png" />

// ✅
<Image src="/assets/image.png" />
```

Reto: sigamos extendiendo el DOM

Reto 1: onLazyLoad

Agregar al componente Image un `prop` llamado `onLazyLoad` que consiste en:

- Una función (callback) cuyo primer argumento es un nodo del DOM tipo imagen, y que no retorna nada (void)

Reto 1: onLazyLoad

```
type LazyImageProps = {
  src: string;
  // Reto:
  onLazyLoad?: // Callback (función)
};
```

Reto 2

El `prop` agregado en el reto anterior – `onLazyLoad` – se ejecutará más de una vez.

- ¿Por qué?
- Soluciónalo. Solo se debería invocar una vez.

Configuraciones Avanzadas

Creando Tipos propios para la aplicación.

Tipos globals

- Idealmente corresponden a **entidades del contexto de la aplicación**. E.g.: **Usuario**, **Producto**.
- No abusar de ellos. Podrían crecer sin control.
- Empieza con tipos locales y expórtalos.

Trabajando con librerías no-tipadas.

Librerías y extensión del objeto Window

Configuración óptima de TypeScript para web