



Svelte Tutorial

- Juan Camilo Flores Vanegas
 - Jonathan Andrés Jiménez Trujillo
 - Diego Alejandro Rodríguez Morales
- 

Índice

1. Qué es un framework?
2. Qué es Svelte?
3. Instalación
4. Estructura en Svelte
5. Svelte vs otros frameworks
6. Particularidades acerca de Svelte
7. *Reactividad*
8. Props
9. Logic
10. Bindings
11. Eventos
12. Lifecycle
13. Conclusión

¿Qué es un framework?

Básicamente es una herramienta que nos ayuda a programar más rápida y fácilmente.



¿Qué es Svelte?

- Es un traductor (compilador) para crear interfaces gráficas para aplicaciones web
- Su lenguaje está compuesto por todos los elementos sintácticos de JavaScript, HTML y CSS.
- Es una solución al problema de construcción de interfaces de usuario usando lenguajes para finalidades diferentes:
 - HTML, CSS: Descripción de forma
 - Javascript: Descripción de comportamiento



Instalación de Svelte

Svelte funciona sobre Node.js, por lo que hay que tener este entorno instalado.

También hace falta Rollup, un Module Bundler. En caso de que no esté instalado se puede instalar usando:

```
npm i -D rollup
```

Para usarlo, se necesita un archivo `rollup.config.js`:

```
export default {  
  input: 'src/main.js',  
  output: {  
    file: 'public/build/bundle.js',  
    format: 'esm',  
    sourcemap: true  
  }  
};
```

Instalación de Svelte

Para usar Svelte con Rollup necesitamos los siguientes plugins:

```
npm i -D svelte rollup-plugin-svelte @rollup/plugin-node-resolve
```

Y especificamos Svelte en el archivo rollup.config.js:

```
import resolve from '@rollup/plugin-node-resolve';
import svelte from 'rollup-plugin-svelte';

export default {
  input: 'src/main.js',
  output: {
    file: 'public/build/bundle.js',
    format: 'esm',
    sourcemap: true
  },
  plugins: [
    resolve(),
    svelte({
      css: result => result.write('public/bundle/bundle.css')
    })
  ]
};
```

Instalación de Svelte

Sólo falta un archivo index.html que utilice el bundle.js generado por Rollup a partir de main.js, y lo mismo pasa con bundle.css.

Entonces, el archivo index.html tendría unas líneas así:

```
<link rel="stylesheet" href="/build/bundle.css">
<script type="module" src="/build/bundle.js"></script>
```

Y el archivo main.js se puede estructurar así:

```
import App from './App.svelte';

new App({
  target: document.body,
  props: {
    name: 'world'
  }
});
```

Instalación de Svelte

Ahora sólo hace falta un archivo App.svelte, que, en un inicio, podría verse así:

```
<script>
```

```
  let name = 'world';
```

```
</script>
```

```
<h1>Hello {name}!</h1>
```

Estructura en Svelte

- Script: Involucrar los elementos necesarios en JavaScript.
- Style: Etiquetas de estilo con respecto al CSS.
- Escribir HTML en el componente sin impedimentos.

```
<script>
  let name = "Juan";
</script>

<style>
  .heading {
    font-size: 3em;
    border: 1px solid #f0f0f0;
    padding: 1.5em;
    color: #23f0c7;
  }
  .about-text {
    font-size: 1.5em;
  }
</style>

<h1 class="heading">Hola bienvenido al blog de Svelte</h1>
<p class="about-text">Saludos {name}</p>
```



Svelte vs otros frameworks

React

Framework desarrollado por Facebook, con una versión inicial en marzo de 2013 y open-source



Pros

- Fácil de aprender y usar
- Basado en componentes: código reusable
- Buen rendimiento y rápido
- Comunidad grande

Cons

- JSX es necesario
- Falta de documentación, no se encuentra completa

Angular

Framework desarrollado por Google con licencia MIT y desarrollada como una plataforma para construir aplicaciones mobile y de escritorio



Pros

- Rendimiento rápido del servidor
- Basado en componentes: código reusable
- Implementación arquitectura MVC
- Documentación buena e intuitiva

Cons

- Curva de aprendizaje pronunciada
- Es demasiado complejo

Vue

Desarrollado por Evan You, con licencia MIT y open-source, es descrito como progresivo, un framework adoptable incrementalmente para desarrollar UI en web

Pros

- Buen rendimiento y rapido
- Basado en componentes: código reusable
- Facil de aprender y usar
- Buena documentacion e intuitiva



Vue.js

Cons

- Pocos recursos comparado con React
- Sobre flexible a veces

Svelte

Framework desarrollado por Rick Harris, open-source y con una primera versión en noviembre de 2016. Es un compilador que coge los componentes declarativos y lo convierte en JS eficiente que se actualiza quirúrgicamente en el DOM



Pros

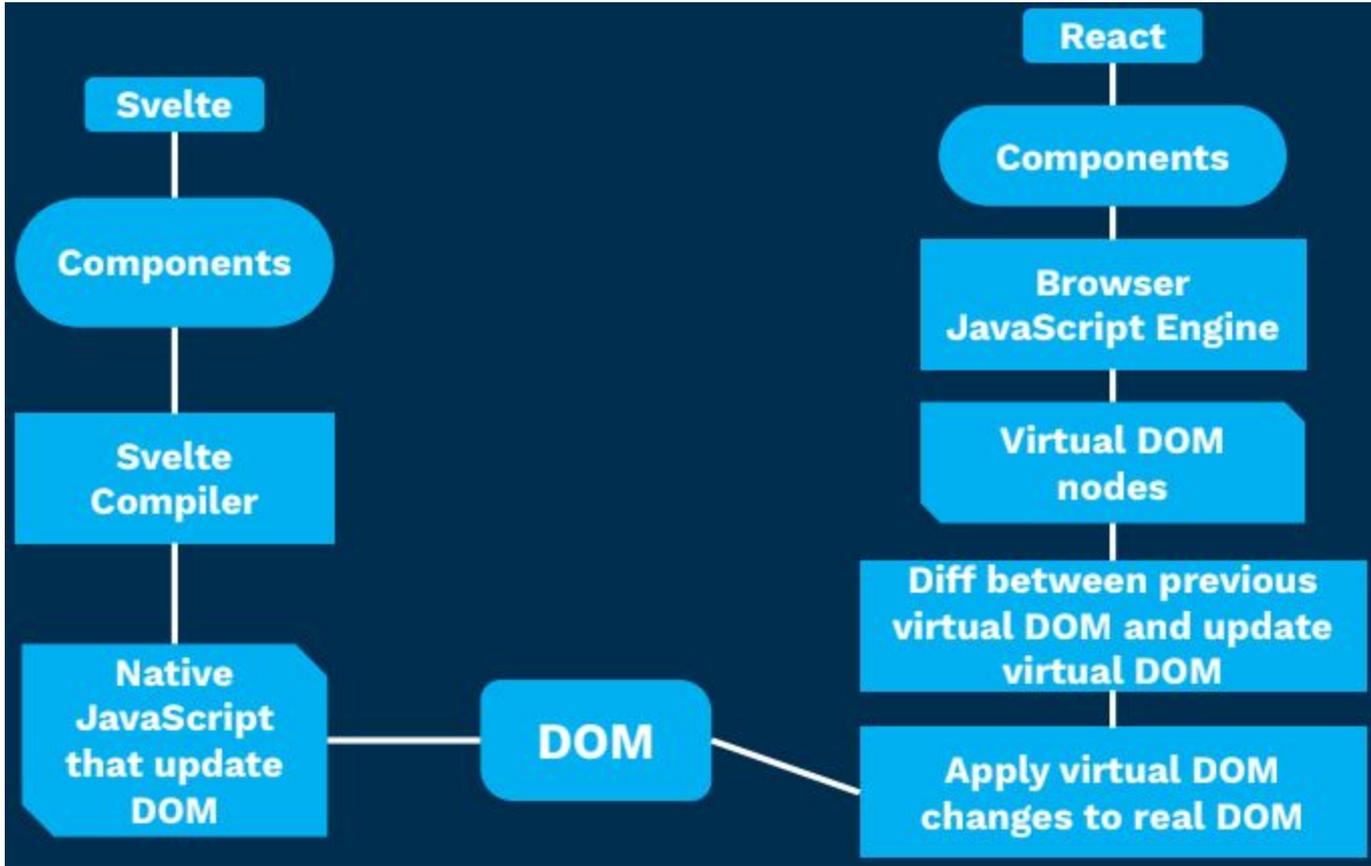
- Sin virtual DOM
- Basado en componentes: código reusable
- Reactivo
- Fácil de aprender y usar

Cons

- Poca comunidad
- Confusión en nombres de variables y sintaxis

Particularidades de Svelte

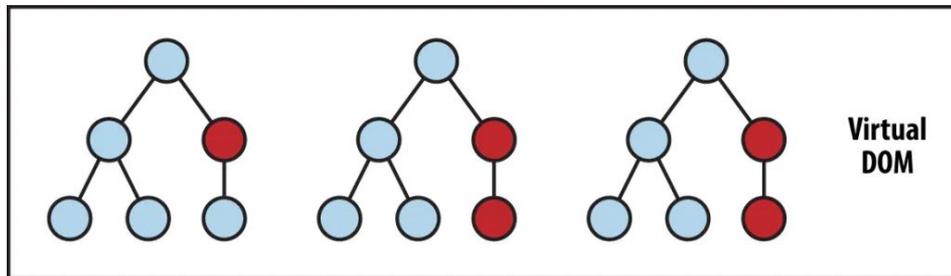
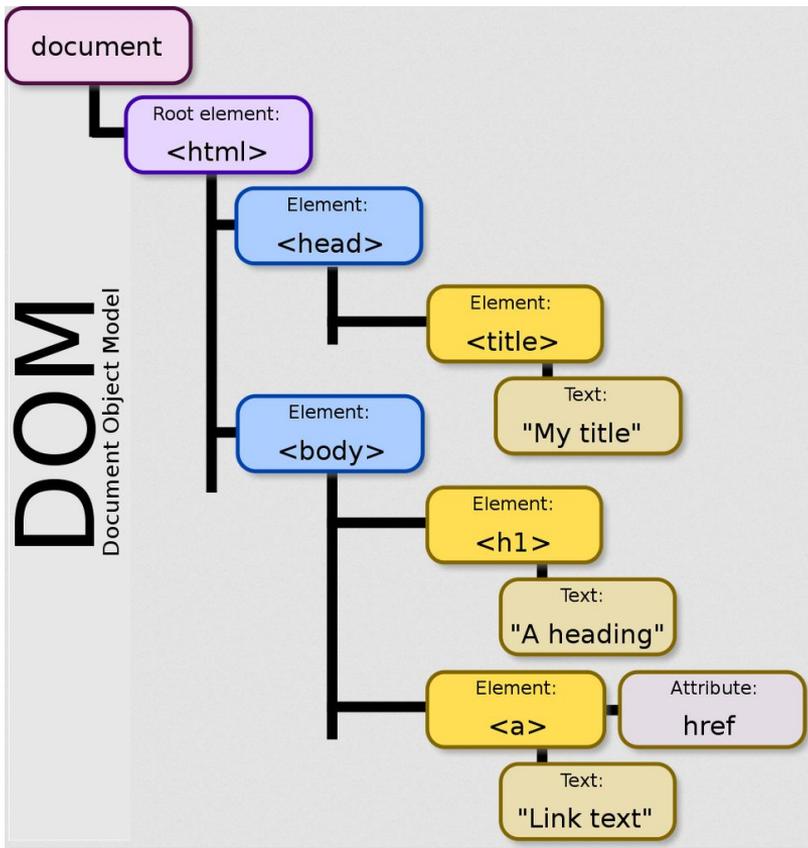
- Tiempo compilado VS Tiempo de ejecución
- Adios al código innecesario
- No utiliza virtual DOM (Modelo de Objetos del Documento)
- Propio manejador de estados (stores)
- Sintaxis más sencilla
- Sapper para SSR y generación de PWA.
- Su API provee muchos recursos con respecto a reactividad, DOM, bindings y el ciclo de vida, entre otros.



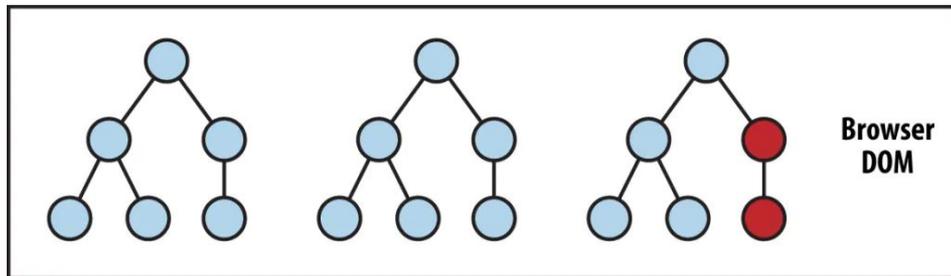


Ejemplos de Svelte

Abstracciones a tener en cuenta



State Change → Compute Diff → Re-render



Conclusión

1. El proceso de traducción de Svelte se compara con la práctica III hecha en clase, en el sentido de que se analiza el lenguaje *Svelte* (que toma como base elementos de lenguajes como JavaScript, HTML y CSS) para producir una salida en JavaScript.
2. *Svelte* es un traductor.
3. Se observa que es una buena herramienta pues facilita muchas tareas al programador, que finalmente se traduce en menores tiempos de desarrollo
4. Lo que logra Svelte en términos de simplicidad semántica (lo que quiero que mi interfaz haga) es lo que ha estado buscando *Vue* o *React*.