

# THE DEVELOPER'S CONFERENCE

Trilha JavaScript

## **Web Components com Vanilla.js**

**Alefe Souza**

# Alefe Souza

Full Stack Developer

@alefesouza

Apaixonado por JavaScript desde os 13 anos, instrutor de desenvolvimento de software, formado em análise e desenvolvimento de sistemas, Microsoft Specialist em tecnologias web.



THE  
DEVELOPER'S  
CONFERENCE



<https://as.dev>

@alefesouza

# Agenda



- O que é Vanilla.js?
- O conceito de componente
- Vantagens
- Live Code
- Compiladores
- Suporte

# Vanilla.js



THE  
DEVELOPER'S  
CONFERENCE

<http://vanilla-js.com/>

Retrieve DOM element by ID

	Code	ops / sec
<i>Vanilla JS</i>	<code>document.getElementById('test-table');</code>	12,137,211
<b>Dojo</b>	<code>dojo.byId('test-table');</code>	5,443,343
<b>Prototype JS</b>	<code>\$('#test-table')</code>	2,940,734
<b>Ext JS</b>	<code>delete Ext.elCache['test-table']; Ext.get('test-table');</code>	997,562
<b>jQuery</b>	<code>\$jq('#test-table');</code>	350,557
<b>YUI</b>	<code>YAHOO.util.Dom.get('test-table');</code>	326,534
<b>MooTools</b>	<code>document.id('test-table');</code>	78,802

Retrieve DOM elements by tag name

	Code	ops / sec
<i>Vanilla JS</i>	<code>document.getElementsByTagName("span");</code>	8,280,893
<b>Prototype JS</b>	<code>Prototype.Selector.select('span', document);</code>	62,872
<b>YUI</b>	<code>YAHOO.util.Dom.getElementsBy(function(){return true;},'span');</code>	48,545
<b>Ext JS</b>	<code>Ext.query('span');</code>	46,915
<b>jQuery</b>	<code>\$jq('span');</code>	19,449
<b>Dojo</b>	<code>dojo.query('span');</code>	10,335
<b>MooTools</b>	<code>Slick.search(document, 'span', new Elements);</code>	5,457

Manipulação de DOM  
AJAX  
Animações  
Bibliotecas de matemática.  
Biblioteca de manipulação de strings e arrays.  
Etc.

@alefesouza

# Vanilla.js

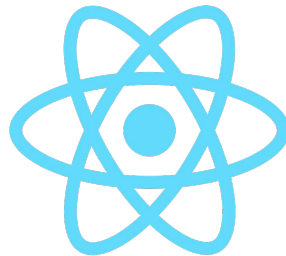


Com tantas bibliotecas e frameworks JavaScript sendo lançadas a cada dia, acabamos não acostumados a utilizar o JavaScript em sua forma mais pura, Vanilla.js nada mais é do que programar em JavaScript sem bibliotecas ou frameworks.

# Componentes



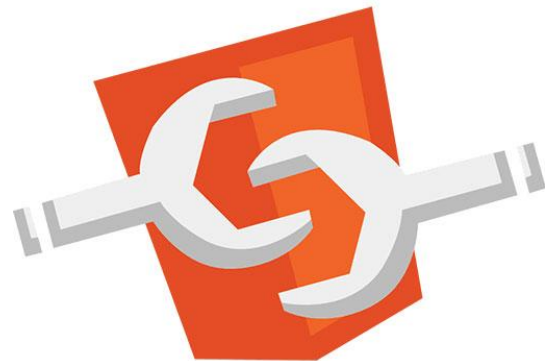
Componentes te ajudam a ter um código mais organizado e reutilizável, também são uma das principais funcionalidades de frameworks modernos.



# Web Components



É uma especificação oficial do W3C que é composta pela união das APIs de Custom Elements, ShadowDOM e HTML Templates.



# Custom Elements



Você consegue ter todo o HTML, CSS e funcionalidade em apenas uma tag HTML customizada utilizando a API de Custom Elements.





# Life Cycle Events



Custom Elements também possuem métodos de life cycle quase iguais aos dos frameworks:

**connectedCallback** - Executado quando o componente é inicializado.

**disconnectedCallback** - Executado quando o componente é removido.

**attributeChangedCallback** - Executado quando um atributo é alterado.

# ShadowDOM



ShadowDOM é uma forma de ter uma conjunto de HTML, CSS e JS separado do DOM principal.

# Vantagens



A principal vantagem de utilizar Web Components é poder escrever seus componentes independente de framework, pois são uma função nativa do navegador.

Se seu projeto é apenas para navegadores modernos, você também não precisa se preocupar com sistemas de build.



THE  
DEVELOPER'S  
CONFERENCE

# Live code!

<https://bit.ly/vanilla-wc>

# Compiladores



O Vanilla.js é muitas vezes substituído por bibliotecas e framework por ser complicado para alguns e às vezes até verboso.

# Stencil



<https://stenciljs.com/>

Ferramenta desenvolvida pelo time do Ionic para facilitar a escrita de custom elements.

Utiliza uma mistura de TypeScript do Angular com JSX do React.

# Stencil



<https://as.dev>

```
import { Component, State, Prop } from '@stencil/core';
```

```
@Component({
  tag: 'my-counter',
  styleUrls: 'my-counter.css',
  shadow: true
})
export class MyCounter {
  @State() contador = 0;
  @Prop() avisar: number;

  increase() {
    this.contador++;
  }

  decrease() {
    this.contador--;
  }

  render() {
    let aviso = null;

    if (this.contador > this.avisar) {
      aviso = <p>0 contador é maior que {this.avisar}</p>
    }

    return <div>
      <p>Contador {this.contador}</p>
      <button onClick={this.increase.bind(this)}>+</button>
      <button onClick={this.decrease.bind(this)}>-</button>

      {aviso}
    </div>;
  }
}
```



THE  
DEVELOPER'S  
CONFERENCE

@alefesouza

# Angular Elements



[https://angular.io/  
guide/elements](https://angular.io/guide/elements)

Angular Elements é um pacote oficial da equipe do Angular que permite empacotar componentes do mesmo em Web Components, dessa forma você pode escrever componentes para qualquer framework utilizando seus conhecimentos em Angular.



# Vue CLI 3



<https://cli.vuejs.org>

O Vue CLI 3 permite a você “buildar” componentes `.vue` como Web Components, assim você também pode escrever componentes para qualquer framework utilizando a sintaxe do `Vue.js`.

# Suporte



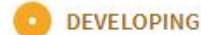
## Browser support



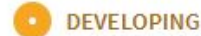
### HTML TEMPLATES



### CUSTOM ELEMENTS



### SHADOW DOM



# webcomponents.org



Esse site funciona como um “armazém” de Web Components, no qual você pode encontrar e até publicar os seus.





THE  
DEVELOPER'S  
CONFERENCE

**Obrigado!**

@alefesouza

<https://bit.ly/vanilla-wc>





# THE DEVELOPER'S CONFERENCE