

JAVASCRIPT COMPLETO ES6+

0 Curso

O Curso

- JavaScript Completo ES6+

var, const, let, class, async, await, fetch, promise e mais

- Pré-requisitos

HTML / CSS

- Ferramentas

Visual Studio Code, Node.js, NPM e Git

Animais Fantásticos

<https://origamid.github.io/animais-fantasticos/>

Grade

- **Introdução**

Sobre a linguagem e configuração o ambiente.

- **JavaScript para Iniciantes**

Dados, variáveis, operadores, boolean, funções, objetos, arrays, loops e mais.

- **DOM para Iniciantes**

Início do projeto, selecionar e manipular o dom, eventos, transversing e mais.

Grade

- **Objetos**

Constructor functions, prototype, String, Array, Math, Number, Function, Object.

- **Efeitos no DOM**

Modules, event bubble, setTimeout, forms e mais.

- **JavaScript Assíncrono**

Promises, Fetch (novo AJAX), JSON, API, HTTP, Async / Await e mais.

Grade

- **Classes**

Class, constructor, get, set, extends e mais.

- **Mais JavaScript**

IIFE, factory function, clojures, debugging, destructuring, rest, iterables.

- **Regular Expression**

Principais padrões e métodos.

Grade

- Automação Front End

CLI, NPM, ESLint, Webpack, Babel e Git.

- Projeto Final

Refatoração do projeto utilizando classes. Praticando com Git, ESLint e mais.

- Slide

Criação do plugin de slide.

Pequeno e Simples

```
const btn = document.querySelector('.btn');

function activeButton() {
  this.classList.add('active');
}

btn.addEventListener('click', activeButton);
```


Pequeno e Complexo

```
export default function fetchBitcoin(url, target) {  
  fetch(url)  
    .then(response => response.json())  
    .then((bitcoin) => {  
      const btcPreco = document.querySelector(target);  
      btcPreco.innerText = (1000 / bitcoin.BRL.sell).toFixed(4);  
    }).catch(erro => console.log(Error(erro)));  
}
```

Grande e Complexo

```
export default class Countdown {  
  constructor(futureDate) {  
    this.futureDate = futureDate;  
  }  
  get _actualDate() {  
    return new Date();  
  }  
  get _futureDate() {  
    return new Date(this.futureDate);  
  }  
  get _timestampDiff() {  
    return this._futureDate.getTime() - this._actualDate.getTime();  
  }  
  get days() {  
    return Math.floor(this._timestampDiff / (24 * 60 * 60 * 1000));  
  }  
  get hours() {  
    return Math.floor(this._timestampDiff / (60 * 60 * 1000));  
  }  
}
```

```
}  
get seconds() {  
    return Math.floor(this._timeStampDiff / 1000);  
}  
get total() {  
    const days = this.days;  
    const hours = this.hours % 24;  
    const minutes = this.minutes % 60;  
    const seconds = this.seconds % 60;  
    return {  
        days,  
        hours,  
        minutes,  
        seconds,  
    };  
}  
}
```

Dúvidas Frequentes

- **Completo ?**

Foco no entendimento completo da linguagem e principais padrões de uso.

- **Vai direto ao ponto?**

Não. O objeto do curso não é ir direto ao ponto, e sim fazer você entender o ponto. Eu rodo o ponto, volto no ponto e demonstro o ponto.

Professor

- André Rafael
- Administrador e Designer
- Mestrando em User Experience Design
- Fundador da Origamid