

# JS Ninja

## ***Como aprender JavaScript do Zero ao Ninja***

Felipe Fontoura

# Conteúdo

Sobre este livro .....	1
Como se tornar o ninja do JavaScript .....	1
Sobre mim .....	1
Depois desse livro .....	2
Contribuições .....	2
JavaScript e o patinho feio .....	3
Instalando os softwares necessários .....	4
Editor de Códigos .....	4
Executando os códigos das aulas .....	4
As demais ferramentas .....	4
Do Básico: Iniciando no JavaScript .....	5
Valores, Tipos e Operadores .....	6
Números .....	6
Operações aritméticas .....	6
Números especiais .....	7
Strings .....	7
Booleanos .....	8
Os operadores de Comparação e Lógicos .....	9
Declarando variáveis .....	10
Primeiros comandos .....	11
Tornando seu programa inteligente .....	12
Criando um robô incansável .....	13
Usando funções para otimizar o seu código .....	14
O Intermediário: Expandindo seus conceitos .....	15
Ao Avançado: Tornando-se um ninja no JavaScript .....	16

# Sobre este livro

Sim! Eu sei o que você está pensando! Mais um livro de JavaScript...

Pois é... A vida de um dev JS não seria completa se:

1. Não criasse mais um *framework* para ver no que dá
2. Escrever um livro
3. Ter um filho (peludos contam)

Sendo assim o #1823466234 livro de JavaScript no mercado serve para te ajudar, já que você terá o exato conteúdo que meus alunos (que pagaram pelo curso) tiveram.

Além de me ajudar a concluir o item 2 da lista.

## Como se tornar o ninja do JavaScript

Eu procurei atender a galera que deseja aprender do zero, por isso você vai ler o livro em 3 partes: o básico, intermediário e avançado.

Sendo assim se você está no zero, recomendo começar pelo começo, devagar. Ou se é mais experiente, sugiro bater o olho nessa parte básica, tirar alguma coisa de útil e seguir para as próximas sessões.

Lembre-se que você não é uma dona de casa assistindo novela das seis, sendo assim, não leia este livro como tal.

A cada capítulo você terá pausas para treinar no seu computador e também ao final de cada capítulo terá desafios.

Na programação não existe fórmula mágica, existe sim é bastante trabalho! A minha parte, que é ensinar, é a mais fácil :)

Para ajudar na fixação do conteúdo, em cada capítulo você terá exercícios práticos e códigos de exemplo que podem ser baixados no repositório <https://github.com/felipefontoura/js-ninja-book>

## Sobre mim

Eu me chamo Felipe Fontoura e sou autor do projeto Dev Samurai (<https://devsamurai.com.br>) e tenho um canal no YouTube (Canal Dev Samurai).

Eu amo programar e estou fazendo isso a pouco mais de 20 anos e nos últimos 5 anos larguei tudo para viver apenas do que eu mais amo **programar** e **ensinar programação**.

Nesse livro eu serei seu professor. Espero que possamos passar ótimas horas aprendendo e se divertindo.

Caso você queira saber mais sobre a minha história, detalhes e também outras coisas que eu curto,

sugiro que leia acesse a minha história em <https://devsamurai.com.br/sobre>

## Depois desse livro

Todo o conteúdo desse livro foi embasado do meu curso online que leva o mesmo nome (JS Ninja).

Caso você acredite que seja interessante e deseje ter algo a mais como:

- Vídeo aulas
- Suporte 1 a 1
- Comunidade de alunos
- Vários outros materiais de bônus

Te convido a experimentar meu curso online JS Ninja sem compromisso por 7 dias. Para isso acesse <https://devsamurai.com.br>

## Contribuições

Caso você goste (ou não) desse livro me envie um *direct* no Instagram para o @devsamurai, ficarei feliz em receber seu *feedback*. Prometo que irei separar a mensagem do mensageiro :)

Caso encontre algum erro ou gostaria de contribuir, abra uma *issue* (correção) no Github, acesse <https://github.com/felipefontoura/js-ninja-book/issues>

De qualquer forma eu espero que você goste do conteúdo do livro e compartilhe com seus amigos que também querem aprender a programar.

# JavaScript e o patinho feio

TODO: Escrever.

# Instalando os softwares necessários

O JavaScript vem instalado por padrão em qualquer navegador e por hora este será o ambiente que iremos passar longas horas se divertindo (querendo cortar os pulsos as vezes).

Então se você já tem o Google Chrome, Microsoft Edge, Firefox, ou qualquer outro navegador atual, pode ficar tranquilo! Você já tem na mão uma das melhores ferramentas de programação.

De qualquer forma nós iremos adotar o Google Chrome como navegador padrão. Se você ainda não tem instalado faça o download pelo site <https://www.google.com/intl/pt-BR/chrome/>

## Editor de Códigos

Para escrever os códigos JavaScript e afins nós iremos utilizar o editor de códigos Visual Studio Code. Para fazer o download e instalar no seu computador acesse <https://code.visualstudio.com/>

## Executando os códigos das aulas

TODO: Google console

## As demais ferramentas

A medida que avançarmos no conteúdo do livro nós iremos precisar de mais algumas ferramentas, como o Node.js, TypeScript, etc.

Por hora, vamos manter os softwares iniciais que são mais que suficiente para começar. E vamos deixar para instalar o restante na hora certa. Fechado!?

# Do Básico: Iniciando no JavaScript

Como qualquer linguagem o JavaScript também implementa o lado básico de qualquer linguagem, que se resume em:

- Valores, tipos e operadores
- Variáveis
- Operadores de comparação e lógicos
- Comandos de entrada e saída
- Estruturas de decisão e repetição
- Funções

Por isso nos próximos capítulos iremos mergulhar fundo nos conceitos básicos para formar uma base sólida e pegar o impulso necessário para os assuntos mais avançados.

Bons estudos!



Caso você nunca tenha tido contato com o mundo da programação os capítulos básicos deste livro serão avançados demais para você, por isso eu recomendo que assista minha aula online e gratuita focada para novatos, acesse <https://programardozero.com.br>

# Valores, Tipos e Operadores

Basicamente o computador é uma máquina de calcular super potente.

Nos primórdios ele foi projetado exclusivamente para realizar cálculos matemáticos complexos. E atualmente além de ser utilizado para jogar e assistir "vídeos educativos", nós programadores utilizamos para criar algoritmos que sejam capazes de resolver problemas reais.

Internamente os computadores modernos implementam todos os números na base binária. Isso significa que o decimal **3** é representado pelo binário **011**.

Isso acontece porque o computador trabalha com Bits (números entre **0** ou **1**).

Para nós, que não somos ninjas ainda, trabalhar diretamente com binário é muito mais difícil. E por isso (e outras razões) que as linguagens implementam a base numérica decimal.

## Números

É um dos tipos de dados mais básicos da linguagem (por razões lógicas).

Os números podem ser Inteiros (*Integer*) ou Fracionados (*Float*). A única diferença importante é que para nós a vírgula representa a divisão de uma fração e para o computador isso é representado pelo ponto.

Portanto, ao representar 3,1 você deve escrever **3.1**. Basta trocar , por .

Nos números muito pequenos ou muito grandes como **1000000000000000000000000** é utilizado a notação científica adicionando um **e** seguido do expoente.

Para o caso acima a representação ficaria como **1e+25**.

No geral o JavaScript não se sai muito bem em computação científica, não foi projetado pra isso. A sua proposta é muito mais voltada ao programador.

## Operações aritméticas

Nenhuma máquina de calcular seria completa sem a **+**, **-**, **\*** e **/**. Que representam soma, subtração, multiplicação e divisão respectivamente.

Estas são as 4 operações básicas e são implementadas diretamente na linguagem.

E como a tia Estela nos ensinou, uma operação de soma, por exemplo, deve ser feita diretamente na linguagem com um simples sinal de "mais": **3 + 3**.

Outro detalhe importante são os parênteses. Eles impõem uma precedência forçada, já que estão no topo da cadeia alimentar:

1. **()**
2. **\* e /**



### 3. + e -

Isso garante que o resultado de  $(21 + 2) * 3$  seja 69 ao invés de 17.

## Números especiais

Na linguagem JavaScript alguns números precisam ser representados de uma forma especial, já que não é muito prático e até possível mostrar na tela o infinito positivo ou o negativo, por exemplo.

Estes dois números especiais são representados para nós humanos na linguagem como `Infinity` e `-Infinity`.

E servem basicamente como resultados para coisas como `1 + Infinity` ou `1 / 0`.

Já para os casos onde o valor não é reconhecido como um número válido nós temos o `NaN` (*Not a Number* em inglês).

Que significa que o resultado ou valor não pode ser utilizado pra nada, sendo apenas informativo, como `0 / 0`, por exemplo.

## Strings

O próximo tipo de dado é a *string*. As Strings são utilizadas para representar um texto e são escritas sempre entre áspas duplas `"` ou simples `'`.

```
"O focinho do babuíno é azul."  
'Desculpe! Já são 3 da madrugada e estou meio confuso.'
```

Ambas as formas de representar uma string estão corretas (com áspas simples e duplas) contanto que as áspas sejam abertas e fechadas da mesma forma.

Qualquer coisa pode ser colocada entre áspas, até os caracteres invisíveis. Explico.

Alguns caracteres como o Enter e Tab, por exemplo, podem ser representados na String adicionando a `\` na frente. Sendo assim o Enter (quebra de linha) ficaria como `\n` e o Tab como `\t`. Veja um exemplo:

```
"O focinho do babuíno é azul.\nE o pelo é cinza!"
```

Como resultado:

```
O focinho do babuíno é azul.  
E o pelo é cinza!
```

As Strings não podem ser multiplicadas, divididas e subtraídas. Contudo podemos realizar "soma"

com o operador `+`.

Com texto, a operação de soma concatena os valores, juntando `'A' + 'B'` para formar `AB`.

## Booleanos

O valor booleano representa o dado mais puro de um computador, já que está associado diretamente a sua base numérica de trabalho.

Um booleano pode ser representado por `1` ou `true` para verdadeiro e `0` ou `false` para falso.

São úteis e mais utilizados em situações onde um dado pode assumir um estado/valor apenas.

# Os operadores de Comparação e Lógicos

TODO: Escrever.

# Declarando variáveis

TODO: Escrever.

# Primeiros comandos

TODO: Escrever.

# Tornando seu programa inteligente

TODO: Escrever.

# Criando um robô incansável

TODO: Escrever.

# Usando funções para otimizar o seu código

TODO: Escrever.



# O Intermediário: Expandindo seus conceitos

TODO: Escrever.

# Ao Avançado: Tornando-se um ninja no JavaScript

TODO: Escrever.