**CURSO DE JavaScript**

CURSO EM VÍDEO

MÓDULO

**A**

AULA 1

O que o JS é capaz de fazer?

Fazendo uma analogia entre um Website e uma casa, o conteúdo em HTML seria a estrutura da casa (fundação, tijolos, cimento etc.), a estilização em CSS seria o acabamento da casa (reboco, tinta, móveis, decoração, janela, porta etc.) e o JAVASCRIPT seria como as interações da casa (a janela e portas abrindo, a tv ligando, a luz acendendo etc.).

O JS é responsável pela responsividade e funcionalidades dos sites onde é aplicado.

AULA 2

JavaScript: Extras

ECMAScript

O ES começou a existir quando os criadores do JS, em 1997, sentiram a necessidade de padronizar a linguagem.

Contrataram a empresa de padronizações, “ECMA”, que lançou sua primeira versão do ECMAScript em 1997. Desde então, lançaram outras versões, seguindo a linha de tempo abaixo:

1.0

2.0

3.0

ES5

ES6

ES

2015

ES

2016

ES

2017

ES

2018

ES

2019

1999

1998

1999

2015

2009

2015

2019

2018

2017

2016

**Tecnologias**

jQuery: Conjunto de bibliotecas, facilita muito o uso de interatividades em JS

Angular: Criado pelo Google, facilita muito a criação de aplicações web. É uma linguagem mais declarativa do que imperativa em si.

React: Criado pelo facebook, tem praticamente a mesma funcionalidade do Angular, mas tem mais flexibilidade. Por exemplo, temos o ReactNative pra criação de aplicações em celular etc.

Vue: Um framework assim como o Angular, mas criado por um ex-programador da Google com a discrição de que “estava cansado do Angular” e resolveu fazer uma versão melhor.

Electron: Mantido pelo GitHub, usado para a criação de programas com Interfaces Gráficas. Por exemplo, o VSCode e versões de alguns apps para desktop como o do wpp, Discord, Slack etc.

Ionic: Um SDK1 para poder criar apps para aparelhos mobile baseado numa versão antiga de uma tecnologia chamada PhoneGap

Cordova: Uma versão evoluída da tecnologia PhoneGap, após ser comprada pela Apache.

1 Kit de Desenvolvimento de Software, um conjunto de ferramentas oferecido por empresas para a criação de aplicativos

**Tecnologias para criar games com JS**

* Phaser
* PixiJS
* Impact
* Melon.js
* CraftyJS

AULA 3

Dando os primeiros passos no JavaScript

**Como aprender?**

Assistindo aula: Assistindo às próprias aulas [deste curso](https://youtube.com/playlist?list=PLntvgXM11X6pi7mW0O4ZmfUI1xDSIbmTm&si=M4PUKz-e_1WvaOFz), mas também procurando tutoriais pela internet. Este curso fornece o conhecimento Básico

Praticando: Acompanhando a aula e colocando a mão na massa! Baixando o software no seu PC e codando!

Lendo: Ler conteúdos, guias de referência, livros**1** etc. Lendo, você consegue aprender de forma mais técnica e dependendo do conteúdo, enxergando outros pontos de vistas sobre a programação

Anotando: Importante anotar os pontos importantes para além de fixar o conteúdo na mente, as anotações tornam mais prática a revisitação dos tópicos

Conversando: Trocar mensagens, visitar fóruns, estudar em conjunto, tudo isso são hábitos que impulsionam a gravação do aprendizado na memória. O ato de tirar dúvidas, ajudar e ser ajudado, é essencial para que você aprenda

Inventar: Aplicar a programação para resolver coisas do seu cotidiano é bem difícil, mas funciona até como um teste, tanto da sua aptidão quanto do seu conhecimento prático.

1Bibliografia

* [JavaScript: O guia definitivo – Flanagan, David (2012)](https://www.mercadolivre.com.br/javascript-o-guia-definitivo-de-flanagan-david-bookman-companhia-editora-ltda-capa-mole-em-portugus-2012/p/MLB19421273?matt_tool=18956390&utm_source=google_shopping&utm_medium=organic&from=gshop#reviews)
* [JavaScript: Guia do Programador – Samy, Maurício (2010)](https://www.amazon.com.br/Javascript-Guia-Programador-Maurício-Silva/dp/8575222481)
* [Guia de Referência(tutoriais) - Mozilla](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript)
* [Guia de Referência ECMA-262 (ECMAScript) - ECMA](https://www.ecma-international.org/publications-and-standards/standards/ecma-262/)

Softwares necessários para usar no curso

Navegador: Vamos precisar de um navegador para acessar a internet e abrir nossos arquivos. Não importa qual o navegador, mas para o curso, instale o [CHROME](https://www.google.com/chrome/?brand=HRJB&brand=HRJB&gclid=CjwKCAjwo9unBhBTEiwAipC1169hieye-fH2eTlawVJOF2pKs5WxX10B3vMiD-Y_xEvtn2uAVf5yfhoCFEIQAvD_BwE&gclsrc=aw.ds).

Editor de Código: Nada mais é que um software de edição de texto. Para o curso, usaremos o [VSCODE](https://code.visualstudio.com/download), que é otimizado para o uso na programação.

NodeJS: Software para a execução do JS fora do navegador, no caso, usaremos para rodar o JS dentro do VSCode. Instale [aqui](https://nodejs.org/en).

AULA 4

Criando o Primeiro Script

Alguns comandos de exemplo

window.alert(): Insere um aviso/mensagem no topo da página, com o conteúdo que estiver entre o parênteses (se for texto, sempre deve-se usar as aspas)

window.confirm(): Insere um aviso/mensagem no topo da página, com o conteúdo que estiver entre o parênteses (se for texto, sempre deve-se usar as aspas). Abaixo deste aviso tem um botão de ok e um de cancelar que funcionam como confirmação e negação.

window.prompt(): Insere uma área de digitação no topo da página. Mostra o conteúdo que estiver entre o parênteses (se for texto, sempre deve-se usar as aspas), como por exemplo uma pergunta.

AULA 5

Variáveis e Tipos de Dados

Comentários

// : Usado para comentar apenas uma linha

/\*

: Comentar mais de uma linha

\*/

Variáveis

Identificador

var nome = “Pedro Judah”

Valor da variável

“Recebe”

Variável

Variáveis: É um espaço na memória do computador destinado a um dado que é pode ou não ser alterado durante a execução do algoritmo. Há 3 tipos:

* var: Usamos para quando o dado pode ser alterado e, uma vez definida, ela pode ser referenciada em todo o documento.
* let: Funciona igualmente a uma var, no entanto, ela só é referenciada no lugar onde for definida (função etc.).
* const: Usamos para quando o dado não pode ser alterado e, uma vez definida, ela pode ser referenciada em todo o documento.

Identificadores: É basicamente o nome dado a uma variável. Pelo identificador que a variável será referenciada no código. Os identificadores devem seguir algumas regras, são elas:

* Podem começar com letra, $ ou \_ (há diferenças entre o uso de cada um)
* É possível usar com letras ou números
* É possível usar acentos e símbolos
* Não podem começar com números
* Não podem conter espaços
* Não podem ser palavras reservadas (ex: “function”, “alert” etc.)

Dicas

* Maiúscula e minúsculas fazem diferença
* Tente escolher nomes coerentes para as variáveis
* Evite se tornar um “programador alfabeto” ou um “programador contador” (“a”, “b”, “c”, “a1”, “a2”, “a3”)

Valor: O valor da variável será preenchido por um dados, para o JavaScript, existem três tipos essenciais de dados. São eles:

* Number: É basicamente composto por números, não importa qual classificação (inteiros, reais etc.), basta apenas possuir caracteres numéricos.
* String: São sequência de caracteres alfanuméricos (números, letras e/ou símbolos) limitados por aspas duplas “”, aspas simples ‘’ e crases ``. (Há diferenças no uso de cada um)
* Boolean: É um tipo de dado que possui apenas duas definições, true e false

Tipos de dados

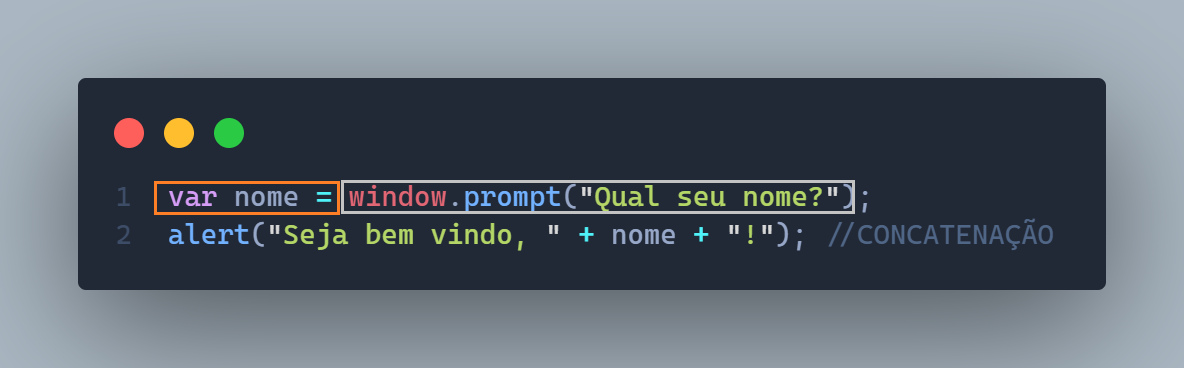
Apesar dos tipos citados acima serem os mais essenciais, existem outros tipos. Abaixo estão listados todos os tipos e em alguns, estão citados alguns subtipos mais comuns. O comando typeof pode ser utilizado para saber qual o tipo do seu dado no JS,

* Number
  + Infinity
  + NaN
* String
* Boolean
* Null
* Undefined
* Object
  + Array
* Function

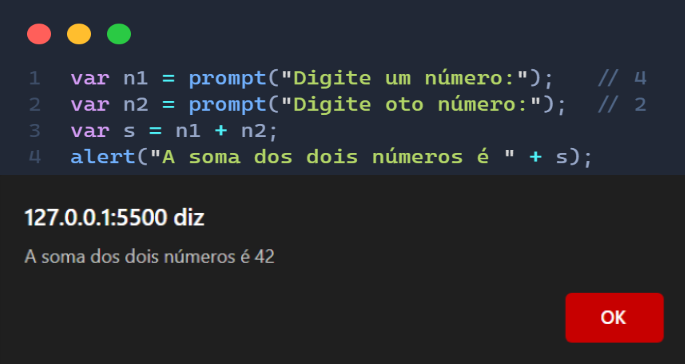
AULA 6

Manipulação de Dados

Guardando os dados

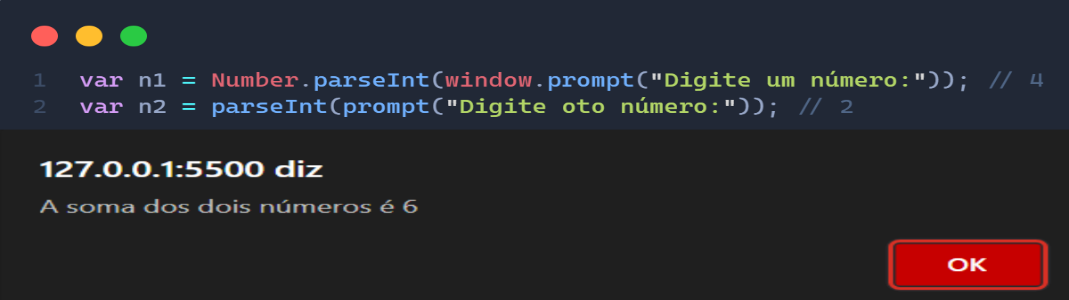
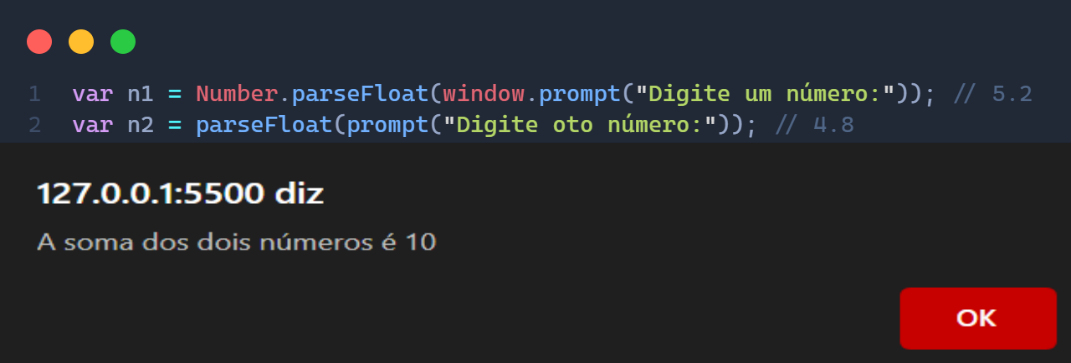
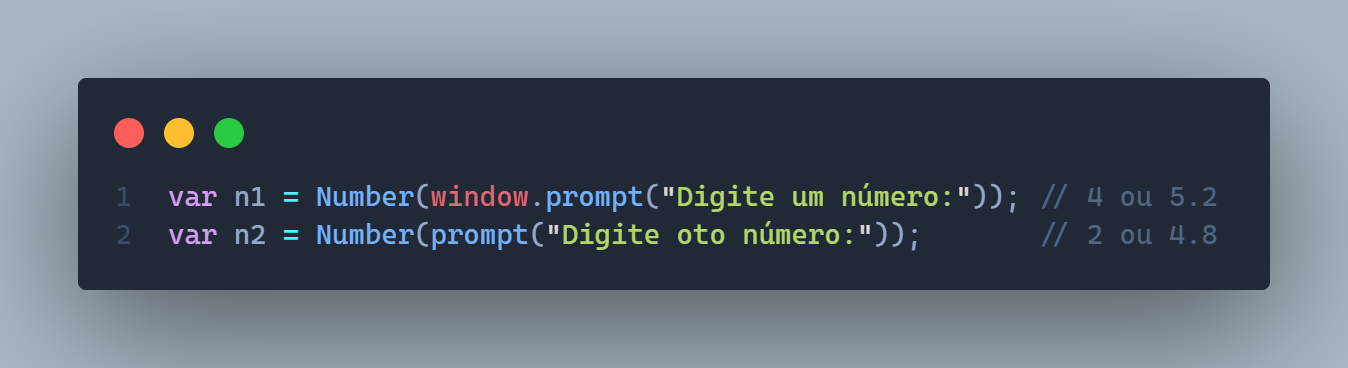
Unindo alguns conhecimentos passados, podemos armazenar os dados que coletarmos por alguns meios (exemplo o comando window.prompt()) em espaços como as variáveis.

Transformação de tipos

O “+” no JS serve tanto para somar, quanto para concatenar. Logo, ao não especificar para o JS, ele fica “na dúvida”. Alguns comandos, por padrão, já definem qual o tipo do dado quando o coleta. No prompt() todo dado inserido é tipado como string (as strings concatenam). Como pode-se notar abaixo:

String 🡪 Number

Para resolver isso, devemos transformar as strings em numbers. Há três formas de se fazer isso:

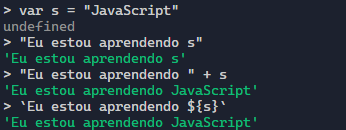
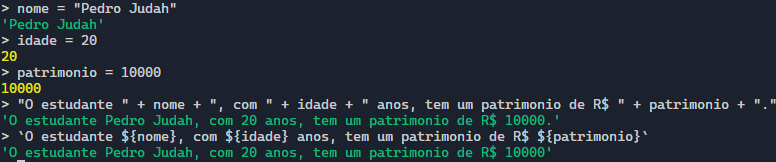
* Number.parseInt(n): Transforma em um número (n) inteiro.
* Number.parseFloat(n): Transforma em um número Real. Chamamos o ponto que separa as casas decimais de “floating point” (ponto flutuante).
* Number(n): Transforma a string em number. O próprio JS identifica se o número é real ou inteiro.

Number 🡪 String

Se necessário, é possível também fazer o caminho inverso, transformar um number em uma string. As duas formas de se fazer isso são as seguintes:

* String(n): Transforma o number em uma string.
* n.toString(): Transforma o elemento que for em uma string.

Formatando Strings

Outra maneira de se mostra os dados é fazendo a interpolação usando as template strings, elas usas as `` ao invés das aspas e ao invés de usar o “+“ para concatenar, usam os ${} (placeholders).

Alguns outros comandos para formatar uma string:

s.lenght: Insere um aviso/mensagem no topo da página, com o conteúdo que estiver entre o parênteses (se for texto, sempre deve-se usar as aspas)