Adicionando o JavaScript a uma página web - Aula 01-02

- Crtl B para recolher aba de arquivos.  
- É possível adicionar o javascript na Tag Head ou Body.  
- Especificar uma tag Script com abertura e fechamento.  
- Linkar o javascript com caminho externo.

O console do browser - Aula 01-03

-DevTools – F12

- Console – Um sandbox para testar o código javascript.

- configurações em 3 pontinhos – settings.

- cosole.log() – método para exibir valores no console DevTools.

Constantes, variáveis e comentários - Aula 01-04

- Váriaveis armazenam algum valor.

- o igual(=) em javascript serve como símbolo de atribuição.

- Dar nome significativos às variáveis.

Tipos de declaração de variável:

Let – variável que permite ser reatribuída.

Cont – Variável constante que não permite ser reatribuída.

Um overview sobre tipos de dados - Aula 02-02

Tipode de dados em JS

Number – Número e decimais no geral.

String – Cadeia de caracteres entre aspas simples ou duplas.

Boolean – Verificar se é verdadeiro ou falso.

Null – Seta, explicitamente, uma variável sem valor.

Undefined – Representa um valor não-setado.

OBject – Estruturas de dados complexas.

Symbol – Utilzado com objetos.

BigInt – Utilizado para manipular números muito grandes.

. Em javascript não é necessário declarar explicitamente o tipo de dado que está armazenando.

Strings - Aula 02-03

. Usado para armazenar letras, números ou qualquer outro tipo de caracterer.

.Concatenação – Nome técnico para juntar uma coisa a outra através do sinal +.

.Acesse um caractere da String através do (nomeDaVariavel[])

.Propriedade length retorna a quantidade de caracteres. Espaços em branco são contados.

.Função é um pedaço de código que executa uma ação específica.

. Um método é uma função embutida no JavaScritp que está associada a um objeto ou tipo de dado particular. São reconhecidas quando há ().

.Alguns métodos não modificam o valor original.

.Alguns métodos esperam por um argumento.

Métodos comuns de strings - Aula 02-04

IndexOf – Procura o caractere que foi passado no argumento.

Slice – Divide a String.

Replace – Substitui caracteres da Strings de acordo com que foi passado no argumento.

Numbers - Aula 02-05

Números decimais tem a notação por ponto e não por vírgula. Exemplo: const pi = 3.14

Ordem de operações

1(primeiro) parênteses

2(segundo) expoentes ou raízes

3(terceiro) multiplicação e divisão

4(quarto) adição e subtração

Atalho para incremento ou decremento: Nomevariavel++ e Nomevariavel—

.Operadores assignment recebem o seu próprio valor e efetuam o calculo.

.NaN – Operação que não resulta em um número.

Template strings - Aula 02-06

.Escapar aspas com \

.Tamplate strings `${}`

.”’’”

Arrays - Aula 03-02

.Array – Armazenar uma lista de valores que geralmente tem uma relação entre si.

.Acessar item através da notação de colchetes.

.Propriedade: length – Tamanho do Array.

.Métodos: Join – retorna uma nova String com todos os itens do array concatenados e separados por vírgula. Ele recebe argumento como separador; indexOf – retorna o índice do argumento; Concat – Pode juntar dois arrays; Push – adiciona ao array; Pop – Retira o último item do array e o retorna. Tanto Push quanto são mutadores de valores.

Null e undefined - Aula 03-03

.Ambos representam a falta de um valor

.Null deve ser intencionalmente atribuído. É usado quando se quer indicar que não existe valor em uma variável.

.Toda vez que não atribuir um valor a uma variável e tenta a usar, o JavaScript vai atribuir a ela como undefined automaticamente.

.Quando se tenta usar undefined em uma expressão numérica é retornado NaN.

.Null é interpretado como zero em expressões numéricas.

Booleans e comparações - Aula 04-02

.Boolena representam dois valores especias em JavaScript: True e False.

.Utilizado para vereficar alguma condição no código.

.Método includes – Verefica se existe a string dentro da variável atribuída como String. Pode ser usado tanto em Strings quanto Arrays.

.JavaScript lê primeiro a letra do alfabeto que vir antes.

.Letra minúscula é maior do que a maiúscula.

.!= == não verefica o tipo do valor.

Comparações com igualdade estrita - Aula 04-03

== =! Comparação não estritos: para comparar dois

valores. Realiza conversão de tipo.

=== ==! Comparação estrita: igual a, e do mesmo tipo(String, number e etc).

Conversão de tipos e 3 avisos - Aula 04-04

.Alterar um tipo de dado para outro através de uma função construtora.

.TypeOf – Vereficar o tipo de um dado.

.Na tentativa de converter uma String em Number é retornado NaN.

Valores Falsy: false; 0; “”, ‘’, ``; null; undefined; NaN.

Valores truthy: Qualquer valor que não é falsy.

.Conversão explícita de dado – Execução de conversão explícita de tipos. Number(), String(), Boolean().

For loop - Aula 01-03

.O propósito de um loop é executar um pedaço do bloco de código repetidas vezes.

.Inicialização – contador: contará quantas vezes o loop foi executado.

.condição: resultará em true ou em false, caso dê false o código do bloco não continuará a ser executado.

.Incremento – Executa sempre no final de cada código, armazenando o valor e adicionando de acordo com a condição.

While loop e exercicios - Aula 01-04

.É necessário declarar no arquivo.

.É preciso lembrar do incremento para não haver um loop infinito.

.i++ dentro do bloco while.

A condicional if - Aula 02-02

.Se a condição resulta em false o bloco de código não executado.

Else e else if statements - Aula 02-03

.Tudo que estiver no bloco de código else será executado se o if resultar em false.

Operadores lógicos && e || - Aula 02-04

.&& - Verificar se duas condições são verdadeiras.

.|| - Apenas uma das condições deve resultar em true.

O operador lógico not (!) - Aula 03-02

.Inverter o Boolean.

Break e continue - Aula 03-03

Break – A iteração dos elementos seguintes do array não acontecerão, pois o execução do loop será parado.

Continue – Pula uma iteração e faz o loop continuar a ser executado.

A condicional switch e exercícios - Aula 03-04

Switch ( ‘Vereficar valor da variável ) {

‘ Declarar série de cases com possiblidades ao valor da variável ’

Break ‘ Faz apenas o case que deu match ser executado e parar o switch’

Default ‘ Ultimo a ser executado, será realizado apenas se não nenhum match com os cases ’

}

O que são funções - Aula 01-02

.Fazem parte do tipo Object.

.Permitem criar um bloco de código em que é invocado e executado quando quiser.

.É possível executa-lo múltiplas vezes.

.É possível passar valores para a função.

Function declaration, function expression e hoisting - Aula 01-03

.Function declaration- Inserção dentro do bloco da function.

.Function expression- Atribuir uma função a uma variável.

.Hosting ( puxar para cima ) – Puxa a declaração da função para o topo do arquivo. Mas não funciona para function expression.

Argumentos, parâmetros e default parameters - Aula 01-04

.Passar valores para dentro das funções.

.argumento – valor; parâmetro – variável.

.A ordem dos argumentos devem cooresponder a ordem dos parâmetros.

.é possível atribuir um valor no próprio parâmetro, mas a função não o executará repetidas vezes.

Retornando valores e exercícios - Aula 01-05

.É possível obter o valor fora do bloco da função retornando o valor ( Devolver um valor para que seja usado fora do bloco dela )

Arrow functions - Aula 02-02

.Forma mais curta e concisa para implementar funções.

.Não precisa usar a palavra chave function.

.Quando há só um parâmetro os parênteses são opcionais.

.Expressão – Qualquer pedaço de código que resulta em um valor.

Funções vs Métodos - Aula 02-03

.O que diferencia um método de uma função é forma de como ele invocado e onde ele é declarado.

.Em um método além do nome e parênteses, precisamos usar a notação de ponto para invoca-la.

.Métodos são funções associadas a objetos ou tipo de dado em si.

O método forEach e callbacks - Aula 02-04

.Callback – Função por argumento.

.ForEach – Método que vai iterar sobre cada item de um array.

O que são objetos - Aula 01-02

.Para elucidar melhor o que são objetos, podemos fazer uma analogia com objetos da vida real, em que há suas características/Propriedades com alguma ação/métodos disponível a ser executada.

.Objetos acoplados como date e math.

Criando um objeto literal - Aula 01-03

.Objeto literais {}

.Dentro do objeto literal podemos colocar propriedades como nome e valor.

.Podemos usar a notação de ponto ou colchetes para acessar alguma propriedade do objeto.

.É possível reatribuir valores com a notação de ponto e colchetes

Adicionando métodos - Aula 01-04

.Propriedade do objeto que armazena uma função se chama método.

.Métodos são funções que são criadas dentro de um objeto.

.Acesso por notação de ponto como um método.

Variáveis e escopo de bloco - Aula 02-02

.Escopo – É a área que um valor de uma variável está disponível.

.Variáveis fora do bloco serão variáveis globais, ou seja, na raiz do arquivo. Ela pode ser acessada ou usada em qualquer lugar do arquivo.

.Podemos criar variáveis de mesmo nome desde que elas não estejam no mesmo escopo.Podemos referencia-los apenas em seu bloco, pois é criado um escopo local.

.A variável referenciar o escopo mais próximo.

.A mesma regras de escopo são aplicadas para a const, mas não podemos reatribui-la.

A palavra-chave this - Aula 02-03

This – Objeto de contexto, ele representa o contexto atual em que o código está sendo apresentado. Ele referencia o objeto em contexto.

.Se quisermos que this o objeto no qual o método foi criado precisaremos usar uma function declaration ao invés da arrow function.

O objeto Math - Aula 03-03

.É um objeto embutido com propriedades e métodos já criados dentro dele.

Tipos de refêrencia vs Tipos primitivos - Aula 03-04

.Tipos primitivos:

.Numbers; Strings; Booleans; Null; Undefined; Symbol e BigInt.

.Armazenados no Stack, uma pilha de diferentes valores que podem ser acessados rapidamente.

.Espaço limitado.

Tipos de referência:

.Todos os tipo de objetos, como os objetos literais, arrays, funções, data e etc.

.Armazenados do heap, tem mais espaço objetos do tipo de referência que são maiores e mais complexos, porém o acesso a ele é mais lento.

.Objetos armazenados no heap são apontados pelo o mesmo ponteiro do Stack.

3 escopos em JavaScript que você deve conhecer - Aula 01-03

.Escopo – A visibilidade de uma variável. Dependendo de onde é declarada outras partes do código podem ou não acessar o valor.

Tipos de escopo:

Escopo de funções- As variáveis declaradas dentro de função serão acessadas somente dentro dela.

Escopo de blocos – bloco de código com anotação de abertura e fechamento de chaves.

Escopo Léxico – Funções aninhadas que buscam variáveis em um escopo acima delas.

O que é o DOM - Document Object Model - Aula 01-06

.É criado pelo o browser. Quando o Html é criado, o browser cria oject model para que seja possível interagir com o documento através do código Javascript.

.Remover e adicionar conteúdo de uma página.

.Document – Objeto que contém propriedades e métodos.

.Descreve a página html em formato de arvore hierárquica ou arvore de nodes.

Query selector & query selector all - Aula 01-07

.Buscar e selecionar um elemento do DOM.

.Busca por tag; por seletor de classe; por referência do seletor da classe e obtendo o seletor único do elemento através do browser.

NodeList – Lista de nós similar a um array, mas não é um array.

.Retornam nodeLists ao invés de html colection.

Adicionando e modificando conteúdo em uma página - Aula 02-02

innertText – The innerText property of the [HTMLElement](https://developer.mozilla.org/en-US/docs/Web/API/HTMLElement) interface represents the rendered text content of a node and its descendants.

.O innerText pega o conteúdo do texto.

innerHtml - The [Element](https://developer.mozilla.org/en-US/docs/Web/API/Element) property innerHTML gets or sets the HTML or XML markup contained within the element.

.Pega o conteúdo e ainda pode modifica-lo.

Obtendo e setando atributos - Aula 02-03

.setAttribute - Sets the value of an attribute on the specified element. If the attribute already exists, the value is updated; otherwise a new attribute is added with the specified name and value.

Modificando estilos CSS (inline) - Aula 02-04

.SetAttribute sobrescreve um elemento já existente ao ser adicionado.

.Style – Podemos usar a propriedade Style para obter, adicionar ou remover estilos de um elemento.

Obtendo, adicionando, removendo e alternando classes CSS - Aula 02-05

.classList – Forma de obter uma listagens das classes que um elemento tem.

.classList.add – Método para adicionar classe a um elemento.

.classList.remove – Remover classe.

Método toggle – Pode adicionar ou remover a classe de um elemento.

Parents, children e siblings - Aula 03-02

Relações dos nós do DOM – Relações de nós diferentes dentro de nós.

Siblings – Quando há elementos no mesmo nível e mesmo pai direto, eles são chamados de irmãos(siblings).

Parents – Usamos esse tipo de relação para atravessar o DOM entre diferentes elementos e obter seleções maiores deles.

**Event bubbling e event delegation - Aula 04-02**

Os eventos são propagados se houver um chamado no elemento pai. O método stopPropagation interrompe esse processo.

eventDelegation – útil ao atrelar eventos em múltiplos elementos.

# Expressões regulares - Aula 01-04

/^string$/ - Dará match apenas se não houver nenhum caractere antes e depois da String.

/^string$/ - É uma lista que dará match com qualquer caractere que for inserido dentro dele.

/^[a-zA-Z]$/ - Expressões maiúsculas e minúsculas.

/^[a-zA-Z]{}$/ - As chaves representam um quantificador que indica a quantidade de caracteres que queremos dar match.

^[a-zA-Z0-9]{6, }$

^.{6,10}$ - O ponto(.) representa qualquer caractere.

Método test()da regex tenta dar um match entre a regex e a string que espercificarmos como argumento da invocação. Ela sempre retorna um boolean.

# Uma introdução ao Bootstrap - Aula 01-02

.Bootstrap é uma biblioteca de componentes front end.

.Componentes são elementos e parte da interface do usuário que podem ser reutilizadas.

# setTimeout() e o objeto window - Aula 02-02

.O objeto global do browser é o window.

.setTimeout(() => {}, ) recebe dois argumentos, uma função e um retorno após um tempo estabelecido.

.ScroolTo()

.O setInterval é parecido setTimeout, a diferença é que a função é executada repetidas vezes de acordo com intervalo e tempo especeficado.

.ClearInterval() Cancela o setInterval.

# O método map - Aula 01-02

.Método de array que pode ser usado para gerar um novo array com a mesma quantidades do original mas cada item transformado.

.Percorre cada item do array, executa alguma transformação em cada um dos itens através da função que passamos como argumento e retorna um novo array com os itens transformados.

.Ele não modifica o array original o qual foi encadeado.

.Todo array tem vários métodos para serem usados, e um deles é o map.

.a função pode receber três parâmetros, o item(o item atual que está sendo iterado), o index(a posição do item que está sendo iterada) e o array(e o próprio array que está sendo iterado). Só o item é obrigatório.

# O método filter - Aula 01-03

.Recebe uma função como argumento e executa essa função para cada item de um array.

.Usada para obter um novo array somente com alguns itens do array original.

.a função pode receber três parâmetros, o item(o item atual que está sendo iterado), o index(a posição do item que está sendo iterada) e o array(e o próprio array que está sendo iterado). Só o item é obrigatório.

.O filter cria um array vereficando quais itens do array original atendem a condição especificada dentro dessa função.

# O método reduce - Aula 01-04

.Recebe uma função como argumento e executa essa função para cada item de um array.

.Usado baseado no array original, é preciso reduzir a algum outro tipo de dado, sendo ele objeto literal, String, um número ou até mesmo um novo array.

.Recebe uma função e um número como argumento.

.O accumulator recebe o segundo parâmetro, o item recebe o primeiro item do array, a função irá retornar o resultado da expressão e reservar no accumulator; e assim sucessivamente até o fim do array.

# O método sort - Aula 02-02

.Ordena objetos, Strings, números e etc.

.Modifica o array original.

.recebe uma função com parâmetros para comparação.

# Data e hora em JavaScript - Aula 01-02

Datas são um tipo de objeto, ou seja, são um tipo de referência e não primitivo.

New é inserido antes de um construtor para que um novo objeto seja criado.

console.log('timestamp:', present.getTime())//milissegundos desde 1900 e bolinhas e é usado para comparação de tempo.

# Timestamps e comparações - Aula 01-03

Para calcular no passado ou futuro podemos passar no parâmetro do construtor data em strings a data desejada.

# Código assíncrono em JavaScript

O que é uma operação assíncrona? É um código que inicia um processo agora e finaçiza esse processo posteriormente.

.O javaScript executa uma instrução por vez, de cima para baixo.

.O javascript é uma linguagem de “único fio”.

.Block code, instrução que bloqueia a execução enquanto faz o request.

# O que são requests HTTP - Aula 01-04

API(Application Programming Interface)

.Requisições HTTP são feitas para se conectar a API ou banco de dados.

.Um requisição/Request é uma solicitação de informações.

.HTTP (Hyper Transfer Protocol) – É o protocolo principal de comunicação de computadores utilizados na internet.

# Fazendo requests HTTP

XMLHttpRequest(): Padrão para troca de informações; objeto utilizado para request do servidor;

.método open() – recebe como argumento uma String com o método request de http e o endpoint http.

Readystatechange – tracker/acessar os dados e saber se eles foram obtidos.

Propriedade readyState – retorna o estado atual da requisição.

Headers\_Received – Headers são cabeçalhos com informações da requisição e são a primeira coisa que recebemos quando enviamos um request através da invocação do send()

.Só no estado 4 que podemos fazer alguma coisa com a resposta.

# Status de respostas HTTP - Aula 02-04

.Status 200 significa que todo o processo ocorreu bem.

.404 – O servidor não conseguu encontrar os dados solicitados por requisição.

# Trabalhando com JSON (JavaScript Object Notation) - Aula 03-04

.Json é uma String que se parece com objeto JavaScript.

.JSON objeto embutido do javaScript com método parse()

.parse() – recebe com argumento uma String JSON e “persea”(converte) para um array de objetos.

.A principal diferença entre declarar uma propriedade Json e JavaScript é que toda as propriedades “chaves” precisam estar dentro de abertura e fechamento de aspas duplas, e quando o valor dessas chaves é uma String, essa String também orecisa dentro de aspas duplas.

# Introdução a promises - Aula 04-03

.Uma promise() é um objeto que representa o sucesso ou uma falha de uma operação assíncrona.

.Uma promise() sempre vai ter dois resultados possíveis resolved, que significa que os dados requisitados foram obtidos, ou reject, que significa que um erro aconteceu e a promise foi rejeitada. A função possui esses dois como parâmetros.

.Método then recebe uma função como argumento que é responsável por receber a resposta de sucesso da promise. O parâmetro value recebe o valor do argumento do resolve.

.Catch trata erros.

# A fetch API - Aula 05-03

.Buscar dados de um outro lugar.

.A fetch retorna uma promise.

.A rejeição é feita somente quando há erro de conexão de rede.

.método .json() permite pegar a resposta que o response está armazenando e parsear a resposta(transformar de String para o objeto).

# Async/await - Aula 05-04

.Syntax Sugar; Uma abstração de Promises.

.O uso da palavra chave async à frente de uma declaração de uma função transforma esse função em assíncrona. Ao incova-la, será retornado uma promise independente do conteúdo dentro dela.

.O uso da palavra chave await pausará o restante da execução da função até que a resposta da invocação do fetch seja obtida.

.Uma função async sempre retorna uma promise que encapsula o valor da função que está retornando. Ou seja, toda uma função async realiza implicitamente uma promise.

# Requests paralelos vs. Sequenciais - Aula 06-03

Para request paralelos, é necessário remover o await antes do fetch; assim, faz com que os fetchs executem as requisições sejam feitas quase ao mesmo tempo e o retorno das promisse sejam armazenadas nas consts.

Para obter o valor das promises podemos usar o método construtor all, ele recebe como argumento um array de promises e quando todas as promises forem recebidas forem resolvidas será retornado uma única promise que contém um array com as promises resolvidas.

# Tratando erros com try/catch - Aula 06-04

Quando um erro é lançado em uma aplicação JavaScript o código abaixo do erro não será executado. Para que o programa não para a partir do erro, podemos usar o try catch que é uma cláusula que tenta executa um código sem travar o restante da aplicação.

# Introdução ao localStorage - Aula 01-03

.É um mecanismo interno com a capacidade de salvar dados no Browser do usuário através de pares de chave e valor.

.2 formas de salvar dados – Banco de dados ou Web Storage API

.Existe a maneira de salvar pelo SessionStorage(Só persiste os dados enquanto durar a sessão)

.IndexedDB só é usado quando é preciso armazenar grande volumes de dados mais complexos.

.Cache API para a aplicação funcionar off-line.

# Armazenando e obtendo dados - Aula 01-04

LocalStorage é um objeto que é uma interface para ser trabalhada, proporcionando uma interação entre o código JavaScript com esse objeto.

.LocalStorage é uma propriedade do Window.

.Um item é um par de chave e valor armazenados na LocalStorage.

.Os dados salvados no LocasStorage serão apagados apenas se forem romovidos ou se o usuário limpar o cache do Browser.

.SetItem() adiciona um chave e valor ou modifica alguma que já existe.

# O que é orientação a objetos

.Orientação a objetos é uma abordagem, uma forma de programar, um paradigma em que dados são encapsulados em objetos e esses objetos são trabalhados no decorrer da aplicação.

.Construtores – São objetos ou funções que criam e constroem novos objetos.

.O “new” é inserido antes da invocação de um construtor para que um novo objeto daquele tipo seja criado.

.O JavaScript usa os construtores por de baixo dos panos para agirem como wrapper objects