{ } Chaves  
[ ] Colchetes  
( ) Parênteses

**1.0 Javascript para iniciantes**

**1.1 Tipos de dados VAR ou LET ou CONST**

Os tipos de constante

**1.2 Tipo Strings**Compondo strings e string Number

**1.2.a Aspas**Aspas duplas em string

**1.2.b Template Strings**`${} ` expressões em strings

**1.3 Números**String de Number

**1.3.a Operadores aritméticos**Operadores matemáticos

**1.3.b NaN**NotANumber  
 **1.3.c Ordem aritmética importa**Ordem das equações matemáticas

**1.3.d incrementando 1**Incrementando com ++

**1.3.e String para Numero**Transformando string em numero

**1.4 Boolean e Condicionais**

**1.4.a Boolean**True ou False

**1.4.b Condicionais if e else**Condicional verificadora if () { } else {}

**1.4.c Condicional else if**

Else if verifica mais de uma vez

**1.4.d Valores que retornam verdadeiro ou falso; Truthy ou falsy**Propriedades que retornam true ou false

**1.4.e Operador logico de negação**Negar operação booleana, if(!true) = se não for falso

**1.4.f Operadores de comparação**Comprando se maior, menor ou igual  
E operadores lógicos como   
&& = e  
|| = ou

**1.5 Switch**Executa uma ação com cada caso escolhido

**1.6 Funções**Básico de funções

**1.6.a Parâmetros e Argumentos**Parâmetros enviado para a função dentro dos parênteses

**1.6.b Argumentos podem ser funções/ funções anônimas**Funções anônimas

**1.6.c Verificando se o dado informado esta correto antes de executar uma função;**Verificar um dado antes de executar a função

**1.6.d Escopo**Variáveis definidas dentro de uma função, não sai dela

**1.7 Objetos**Variáveis com diversas propriedades

**1.7.a Metodos – functions / função em objetos**Adicionando uma função em uma propriedade

**1.7.b Acessando a propriedade e atribuindo valor**Acessando atribuindo valor a propriedade de um objeto

**1.7.c Incluindo propriedade a um objeto já criado**Incluir propriedade no objeto

**1.7.d this**Dentro de um objeto, o this é utilizado para mencionar a propriedade em questão

**1.7.e Tudo é objeto**Tudo tem métodos e propriedades, portanto é um objeto

**1.7.f QuerySelector**Selecionando elementos com Query Selector

**1.8 Array**  
Tipo de objeto com índice e próprios métodos e propriedades

**1.9 Loop – for**Looping for, var contadora recebe valor, enquanto for diferente, recebe valor

**1.9.a Executando for com array**Utilizando for para varrear a Array

**2.0 While loop**Repetições de while/enquanto

**2.1 Break**Parar a repetição quando achar dado especifico

**2.2 forEach**Realiza varredura em toda Array

**2.3 Operadores de atribuição**Atribuindo valores realizando euqções

**2.3.a Operador ternário / Abreviação if e else**Atribuir valores a partir de verificações

**2.4 Escopo const e let**var vaza o bloco de function, por isso é utilizado const e let

**2.4.a Const e Let**Definições de const e let

**3.0 Dom para iniciantes  
3.1 DOCUMENT OBJECT MODEL (DOM)**Dom representa toda a janela, ao colocar um elemento em uma constante, ela tem as propriedades de html

**3.1.a addEventListner – Função anonima**Adicionando método com função anonima

**3.1.b addEventListner – função separada**Adicionando método com função separada, com callback

**3.2 Seleção de elementos**

**3.2.a GetElementsById – único**selecionar um elemento por Id

**3.2.b GetElementsByClassname - Retorna HTML Collection**Selecionar um grupo de elementos por classe

**3.2.c GetElementsByTagName - Retorna HTML Collection**Selecionar um grupo de elementos por name

**3.2.d QuerySelector – único**Seleciona o primeiro elemento que corresponder **3.2.e QuerySelectorAll - Retorna NODE list**Seleciona vários elementos de acordo com parametro

**3.2.f HTMLCOLLECTION vs NODELIST**Tem diferentes propriedades e métodos, e HTML Coleection atualiza em tempo real, Nodelist não

**3.2.g Array-like – foreach – Array from**Transformar HTML Coleection em Array

**3.2.h Arrow Function**Encurtando function com Arrow Function

**3.2.i – Arrow Function Reduzido**Caso retorne apenas um valor, podemos reduzir o Arrow Function

**3.3 Adicionar remover ou alterar classes**Adicionando, removendo ou alternando classes

**3.4 Attributes**Atributo dos elementos como id e class

**3.4.a getAttribute setAttribute**Adicionando, removendo ou verificando atributos de um elemento

**3.4.b READ ONLY VS WRITABLE**Algumas propriedades não aceitam mudanças

**3.5 Dimenções e distancia**distancias e dimensões do cliente

**3.5.a Height e width**Altura e largura da pagina do cliente

**3.5.b offsetTop e offsetLeft**Medir distancia de um objeto até o topo ou a esquerda de um objeto

**3.5.c getBoundingClientRect()**Verifica todas as coordenadas de um elemento

**3.5.d Window**Medidas da janela

**3.5.e MatchMedia() – Verifica por JS a largura do navegador**

Combinando medias a medidas de janela

**3.6 Eventos**eventos que ocorrem em ações como o clique

**3.6.a addEventListener**Adiciona função ao elemento, quando ocorre é ativado o callback

**3.6.b addEventListener > function > event**Event é o argumento passado na function de callback

**3.6.c event.preventDefault()**Prevenindo o evento padrão

**3.6.d This em addElementListener**This faz referencia ao elemento que chamou a função

**3.6.e diferentes eventos de event listener;**Tipos de evento que realiza o callback

**3.6.f Foreach em eventlistener**uma function de callback pode realizar o foreach dentro dela

**3.7 Traversing e Manipulação**

**3.7.a OuterHTML – InnerHTML e InnerTEXT**Interagindo com todo html

**3.7.b TRASVERSING** Navegando pelos elementos do DOM

**3.7.c Element vs Node**PreviousElementSibling = element anterior  
PreviousSibling = Qualquer item anterior

**3.7.d Manipulando / movendo elementos**Adicionar, remover ou substituir um elemento

**3.7.e Novos elementos – createElement()**Criando elementos

**3.7.f Clonando elementos**Clonar um elemento com outra const

**3.8 Navegação por tabs**Tab de navegação que exibe uma lista conforme item clicado, utilizando foreach e event listener

**3.8.a Só ocorre caso tenha Javascript**Adiciona uma classe utilizando Javascript, assim o CSS de JS só ocorre caso a pag tenha JS

**3.9 Acordeon list**Lista com tab de informações oculta que aparece ao click

**4.0 Scroll Suave**Link interno rola para o destino suave

**5.0 Objetos**

**5.1 Constructor Function – Função Construtora** Função com uma base para criar objetos semelhantes, podendo ter metodos

**5.2 Prototype**Propriedade que todas as funções tem, é capaz de criar novos métodos para funções construtoras

**5.2.a Entendendo o que está retornando;**Entenda qual o tipo

**5.2.b Verificando os métodos**Verifica quais métodos podemos utilizar com um dado especifico

**5.3 Objects - Native, host e user**objetos do JS, navegador ou usuario

**5.4 Metodos e propriedades da STRING**

**5.4.a Length**Quantidade de caractere

**5.4.b Charat**Encontra letra em determinada posição

**5.4.c Contatenando Strings**Juntar duas strings

**5.4.d Includes**Verificar se a string possui um dado

**5.4.e Startswith Endswith**Verificando se a string começa ou termina com algum dado

**5.4.f Slice**Divide a String

**5.4.g Indexof e Lastindexof**Retorna o index do caractere passado

**5.4.h Padstart**Determina tamanho da string, completando com caractere informado

**5.4.i Repeat**Repete a string

**5.4.j Replace**Substitui determinado caractere por outro passado

**5.4.k Split e Join**Divide transformando em uma Array, e junta adicionando um dado passado

**5.4.l toUpperCase e toLowerCase**Para maiúscula ou minuscula

**5.4.m Trim**Remove espaços do começo ou fim

**5.5 Number**

**5.5.a verificar se é numero isNaN**Verifica se não é um numero

**5.5.b isInteger verifica se é integral**Verifica se tem virgula

**5.5.c ParseFloat**Transforma a string em numero caso ela começe com um numero

**5.5.d ParseInt**Transforma um numero quebrado em integral

**5.5.e toFixed**Arredonda com parâmetro de casa decimal

**5.5.f toString**Transforma Number em String

**5.5g LocalString**Number para string de moeda indicada

**5.6 Math**Objeto nativo com diversas propriedades

**5.6.a Arredondamento de números**Arredondamento e para positivo

**5.6.b Min e max**Retorna o valor mínimo de uma array

**5.6.c Gera numero aleatório**Pode ser gerado entre dois numeros

**5.6.d tres pontos para verificar um math**É utilizado 3 pontos para a propriedade math verificar toda a array

**6.7 Array**Armazena diversos dados, podendo ser string, number, boolean, function, object, etc

**6.7.a Construção de Arrays**Construção básica de uma Array

**6.7.b Array From – transforma em array**Transforma uma Array like em Array

**6.7.c IsArray – verifica se é array**Verifica se é Array

**6.7.d Criação de array - Array.of Array() e New Array()**Criar uma Array com base em dados, ou uma array vazia

**6.7.e Length – tamanho da array**Retorna o tamanho da propriedade da array, ou do objeto dentro dela

**6.7.f Sort**Organiza a Array em ordem

**6.7.g Unshift e Push – adicionando e removendo elementos**Adiciona propriedades no inicio ou no fim da Array

**6.7.h Shift e Pop Remove itens**Recebe e remove ou apenas remove item da Array

**6.7.i Reverse – invertendo a ordem**Apenas inverte a ordem da Array

**6.7.j Splice adicionando em index**Adiciona propriedade no index passado

**6.7.k Copy within**Copia uma parte da Array

**6.7.l Fill() – preencher**Preenche ou substitui valores da Array

**6.7.m Concat – Contatenar**Junta duas arrays

**6.7.n Inclues, Indexof e Lastindexof**Verificando se a array tem, ou qual o index, ou retornar o ultimo

**6.7.o Join - Juntar valores Split – Separar**Mesmo split e Join de string

**6.7.p Split separa**Mesmo split e Join de string

**6.7.q Slice Corta pedaço**corta a Array

**6.7.r clonar array**Cola em uma nova Array parte de outra

**6.8 Metodos de interação objetos Array**Diferença entre array e map

**6.8.a Diferenças entre Array e MAP**Array apenas modifica, Map guarda em uma variavel

**6.8.b ForEach**ParaCada item da Array é executado

**6.8.c Modificando a array original com o foreach**É possível modificar cada item da Array com forEach

**6.8.d Arrow Function**Abreviação de function

**6.8.e Map**Para modificar os valores de uma array utilizar o map

**6.8.f Criando uma const com uma função embutida**Const com uma Function

**6.9 Reduce – Acumulador**Funciona como o ForEach, mas tem uma propriedade que acumula valor, podendo somar toda uma Array

**6.9.a Reduce Equações**Soma

**6.9.b Reduce maior ou menor numero**Verificar maior ou menor numero de uma Array

**6.9.c Reduce passo a passo**Como funciona

**6.9.d Retornando propriedades de objetos com reduce**Exemplo passando a propriedade para uma const, limpando e somando no acumulador

**6.9.e Reduce Right**Retorna invertido

**7.0 Some**Funciona como o forEach, caso uma afirmação for verdadeira, retorna true

**7.1 Every**Funciona como o forEach, caso todas afirmações for verdadeira, retorna true

**7.2 Find e Find Index**

**7.2.a Find**Varre a Array e retorna o que for verdadeiro, como por exemplo números maiores que 45

**7.2.b FindIndex**Retorna apenas o primeiro valor que for truthy

**7.3 Filter**Filtra toda a Array e retorna todos valores truthy

**7.3.a Filter em Objetos**Busca objeto dentro do objeto, a propriedade especifica

**7.4 Function**

**7.4.a Propriedades**Podemos utilizar propriedades em function, como length

**7.4.b Call**Aciona uma função passando objetos diferentes (mudando o this)  
Exemplo criando um seletor de elemento, e um adicionador de classe,   
E utilizando o call posteriormente para adicionar uma classe utilizando o seletor e o add.

**7.4.c Apply**Funciona como o call, mas passa os argumentos em array

**7.4.d Bind**A partir de uma nova const, utiliza a function com um objeto  
Ou em uma função construtora, a partir de uma nova const utiliza a function com argumentos

**7.5 Object**Todo objeto é criado com o constructor Object, por isso herda suas propriedades

**7.5.a Métodos de Object - Objetos base com tipo protótipos**Exemplo de um objeto construtor com funções  
Ao criar outra constrante com parâmetros Object.create criamos um novo objeto

**7.5.b Object Asign**Adicionando métodos ao prototype de objetos

**7.5.c DefineProperties** Adicionando propriedades imutaveis

**7.5.d Get e Set**Get- Função dentro do objeto capaz de realizar calculo para atribuir valor com base em parâmetro passado na call  
Set- Seta o valor a ser passado

**7.5.e GetOwnPropertyDescriptors**Verifica metodos e propriedades de um dado

**7.5.f Object keys - values – entries**Pegar valores de um objeto

**7.5.g Object GetOwnPropertyNames**Puxa valores das propriedades de um objeto

**7.5.h Object GetPrototypeOf e Objects**Verifica se dois objetos são os mesmos

**7.5.i Object freeze seal e prevenExtension** Impede um objeto de alteração

**7.6 Propriedades e métodos do protótipo**Verificando o construtor de um objeto

**7.6.a asOwnProperty**Verificando se objeto tem propriedade

**7.6.b PropertyIsEnumerable**Verifica se é enumerable

**7.6.c IsPrototyoeOf(Valor)**Verifica se é prototype do valor passado

**7.6.d To String**Transforma em string

**8.0 Efeitos no DOM**Todo elemento HTML do DOM herda propriedades e métodos do construtor HTMLElement.

**8.0.a Dataset**Elementos com marcador data- no HTML vai para JS como objeto  
Para realizar ajustes no CSS utiliza CamelCase  
data-Também vai para JS com CamelCase

**8.1 Modules**Dividindo operações de JS em arquivos separados

**8.1.a Exportando valores**Exportando valores para outro arquivo JS

**8.1.b use strict**Previne erros, Module já tem

**8.2 Janela Suspensa**Criaçao de model, janela suspensa

**8.3 Animação ao Scroll**Animação lateral ao scroll da pagina

**8.4 Tolltip**Caixa de informação que segue o mouse

**8.5 Objetos como funções**Utilizando objetos com funções

**8.6 Dowp down menu**Dropdown menu, subopção de menu

**8.7 SetTimeOut**Intervalo até uma ação ser executada

**8.7.a Metodo com contagem regressiva;**Contagem regressiva até ação

**8.7.b SetTimeOut pertence ao window**Utilizar sempre com event listener + arrowfunction

**8.7.c Set interval**Ação de tempo em tempo

**8.7.d clear interval**Ação de tempo em tempo até que

**8.7.e Ex. de interval**Exemplo de timer

**8.8 Menu mobile**Menu mobile – Menu hamburguer

**8.9 – observador**Observa um elemento até ter mudanças

**9.0 Animação em números**Anima uma contagem de numeros

**9.1 Data Object**Objeto com propriedades de tempo

**9.1.a Tempo até**Contagem dias até

**9.1.b Horario de funcionamento**Verifica se o dia e hora está dentro do informado

**9.2 Formularios**Puxando dados

**9.2.a Values**Puxando valor ao change

**9.2.b Validação**com CheckValidity informamos mensagem de retorno  
Um retorno visual de dados incorretos

**9.2.c Select Checkbox e Radio**Adicionando o valor de value para o css

**9.2.d Recebendo os dados em um objeto**Adicionando todos os dados do formulário em um objeto

**9.3 Button Creator**Objeto com um elemento, e vários métodos que muda o CSS

**9.4 SetValues**Salvar dados do usuário dentro do navegador, com localStorage

**10.0 JavaScript Assíncrono**

Sincrono segue para a próxima quando acabar, Assincrono Executa a tarefa no fundo

**10.1 Promisses**Promise é uma função construtora de promessas.

**10.1.a Promises – Then**Com o metodo then, a função só é executada quando a promessa for true **10.1.b Promises – Catch**Podemos realizar algo caso a função for false com o catch

**10.1.c Promise Finally**O Finally executa algo ao fim da função

**10.1.d Promise all**Quando um grupo de promises forem finalizados, é execuado o .all

**10.1.e Promise Race**Executa quando a primeira promise for resolvida

**10.2 Fetch API**API = interface de aplicação  
Fetch puxa algo, sendo uma api ou arquivo

**10.2.a Json e then**JSON é um arquivo de texto, que pode se tornar um objeto

**10.2.b Blob**Recebendo o url de uma imagem

**10.2.c Clone**Para transformar text em json precisamos clonar

**10.2.d Headers**Informações de cabeçalho

**10.2.e Status**Verifica o status do fetch

**10.2.f URL**Informa o URL do fetch

**10.2.g Type**Verifica o tipo o fetch

**10.2.h Exercicios fetch**Api de cep – Atualização e tempo real – Api ao clique

**10.3 Jason 2**Composição do JSON

**10.3.a Criando Array com objetos**JSON como uma array de objetos

**10.3.b Metodo Parse para transformar text em objeto**Transformando text em objeto com parse

**10.3.c Transformando objeto em arquivo json**Transformando um objeto JS em json com stringfy

**10.3.d Enviando objeto para o local storage como json**Salvando todo um objeto no LocalStorage

**10.3.e voltando como objeto utilizando o parse**Retornando do localStorage como objeto

**10.4 API**application programming interface

**10.5 HTTP**Hypertext Transfer Protocol – recebendo api da internet

**10.6 URL E METHOD**Recebemos por url com o get

**10.6.a GET**Padrão - Recebe

**10.6.b POST**Envia para o servidor

**10.6.c PUT**Atualiza o conteúdo do servidor

**10.6.d HEAD**Puxa o cabeçalho

**10.6.e Cors**Restrição de uso de sites como apis

**10.7 Async Await**Os dois comandos faz a função parar e esperar pelos dados

**10.7.a Then e Async**Async com await exclui a necessidade de uso do then, o que torna a sintaxe mais limpa

**10.7.b Try e Catch em async**Lidando com erros das promises executando algo com catch caso o try não funcionar

**10.7.c iniciando fetch ao mesmo tempo**Podemos realizar vários fatch ao mesmo tempo, mas a resposta deve aguardar o fetch.

**10.7.d o resultando a frente do await deve ser uma promisse, caso não seja o await não tem efeito**A frente de await devemos ter uma promisse que da result

**10.8 History API**Acessando o histórico de navegação so site

**10.8.a PushState**Atualizando o url com pushState

**10.8.b PopState** Executa função ao clicar em voltar ou avançar

**10.8.c FETCH E HISTORY**Simulando navegação com fetch

**10.9 Trocando Html da pag atual**Substituição total ou parcial do conteúdo da pagina

**11.1 inserindo informações do json no html**realizando fetch com await de uma api com dados e inserindo no HTML

**11.2.a Api puxando valor de bitcoin**Realizando fetch com then e inserindo no html

**11.0 Classes**Função construtora class

**11.0.a Constructor function**Função construtora comum

**11.0.b Class**Função construtora com novo método ES6

**11.0.c Constructor**

Parametros agora são passados dentro do constructor

**11.0.d Exemplo Função construtora class**Exemplo de construtora em class

**11.0.e objeto em uma propriedade**Objeto como propriedade de uma class

**11.0.f Class com static**Metodo Static para criar métodos dentro de uma class

**11.0.g Static – Retornando pré definições da class**Pré-definindo valores com static

**11.1 Get e Set em class**Metodos get e set

**11.1.a Get – valor estatico**Get – Informação estatica

**11.1.b Set – setar valores**Set – setar novo valor

**11.1.c Class e get set**Pré-definindo valor com get e set

**11.1.d Class extends**Criando extenção da class

**11.1.e Class extends Super**Utilizando o método Super para executar a função da class + o extends

**11.1.f Class extends Super propriedades**Utilizando o super para extender as propriedades da class dentro do extends

**11.2 Plugin Count Down**Class com metodos get que recebem e realizam contagem regressiva de tempo e retorna

**12.0 Mais JavaScript**

**12.1 Function declaration**Declaração padrão

**12.2.a Hoisting**Diferença na ordem de declaração da função

**12.2.b Arrow Function – expression.**Arrow = expression

**12.2.c Estrutura / Preferencia**Expression = Function dentro de uma const, necessita ter a chamada após a declaração  
Declaration = function declarada, é executada mesmo se a chamada for antes  
  
**12.2.d IIFE - IMMEDIATELY INVOKED FUNCTION EXPRESSION**Metodo utilizado antes do escopo

**12.3 Factory Function**Funções construtoras, que retorna propriedades e metodos

**12.3.a Metodos e variáveis privadas**Podemos declarar métodos e propriedades dentro da factory function que não vasa o escopo pois são privadas

**12.3.b Congelando o retorno**Congelando o parâmetro de retorno

**12.3.c Constructor function Sem o new**Declarando uma constructor function sem o new

**12.3.d Uso real factory function**factory function com diversos metodos de interação com elementos

**12.4 Escopo e closures**Como funciona o escopo de uma function

**12.5 Debugger**Debugando no chrome

**12.6 Destructuring**Cria uma nova const com o dado de um objeto

**12.6.a Destructuring objects**Retirando informação de objetos dentro de objetos

**12.6.b Nesting**Aninhando uma desestruturação, dentro de outra

**12.6.c DESTRUCTURING ARRAYS**Destructuring uma Array

**12.6.d Declarações múltiplas de variável**Declarando varias variáveis de uma vez, também serve para inverter o valor de duas variaveis

**12.6.e Destructuring de objetos**Desestruturando propriedade de um objeto

**12.6.e getComputedStyles**Desestruturando as propriedades de dados computados, como style

**12.7 Parametros**Parametros pré definidos

**12.7.a Parametro inicial – padrão**Caso um parâmetro não seja definido, podemos pré definir um valor

**12.7.b Arguments**Podemos puxar os argumentos com o método arguments

**12.7.c parâmetro rest – lista de argumentos**Criando uma array de argumentos

**12.7.d Operador Spread**Utilizando spread utilizamos variáveis já criadas para interar outra

**12.7.e spread argument**Podemos utilizar o argument por exemplo para passar uma lista de números por um math

**12.7.f transformando em array com o spread**Array.from com spread

**12.8 Iterable**Objetos que tem Symbol.iterator

**12.8.a For of**Looping com for of

**12.8.b For in**Looping em objetos utilizando o for in

**12.8.c for in chave e valor**Utilizando o spread na propriedade style de um elemento e o for in para exibi-la

**13.0 Regular Expression**Utilizando para busca e substituições

**13.0.a Literal**Seleciona toda a frase

**13.0.b Flag: g**Altera a seleção para por letra

**13.0.c Flag g + i**Tag i não diferencia maiúsculas de minúsculas

**13.0.d Character class []**Colocando as letras entre [ ] selecionara cada uma

**13.0.e Characteres especiais**Colocando dentro de [] também podemos procurar caracteres especiais

**13.0.f um ou outro**Procurar uma frase que seja de um jeito ou de outro

**13.0.g buscando de a a z**A a Z ou 0 a 9,

**13.0.h Negar**[^] Negar

**13.0.i Caractere especial ponto**ponto seleciona tudo, menos quebra de linha

**13.0.j Escapar caracteres especiais**utilizando a barra invertida, escapamos caracteres especiais

**13.0.k Word**\w irá selecionar todo caractere que não seja especial

**13.0.l Not Word**\W irá selecionar tudo que for especial

**13.0.m Digit**\d irá selecionar todos os digitos

**13.0.n Not Digit**\D irá selecionar todos que não forem digitos

**13.0.o White space**\s irá selecionar todos os espaços em branco, incluindo tab e quebra de linha

**13.0.p Not White space**\S seleciona tudo que não for espaço em branco

**13.0.q Quantificador min e max**Selecionar quando uma letra se repetir quantidade indicada

**13.0.r Mais +**Seleciona quando se repetir uma vez ou mais

**13.0.s Asterisco \***Seleciona quando exitir com 0 ou mais ocorrencias

**13.0.t Opcional ?**Infororma quando um caractere for opcional

**13.0.u Alternado |**Seleciona um ou outro

**1.3.0.v Word Boundary**Informa quando a palavra terminar com um não caractere

**13.o.x Anchor beginning ^**informa que a busca deve começar no inicio da linha

**13.o.y Anchor End**informa que a busca deve começar no fim da linha

**13.o.z Flag M**Começo ou final da linha

**13.o.a2 line feed /n**Seleciona o final da linha

**13.0.b2 Unicode**Seleciona caractere especifico unicode